computing today

ISSN 0142-7210
NOV 1979
50p

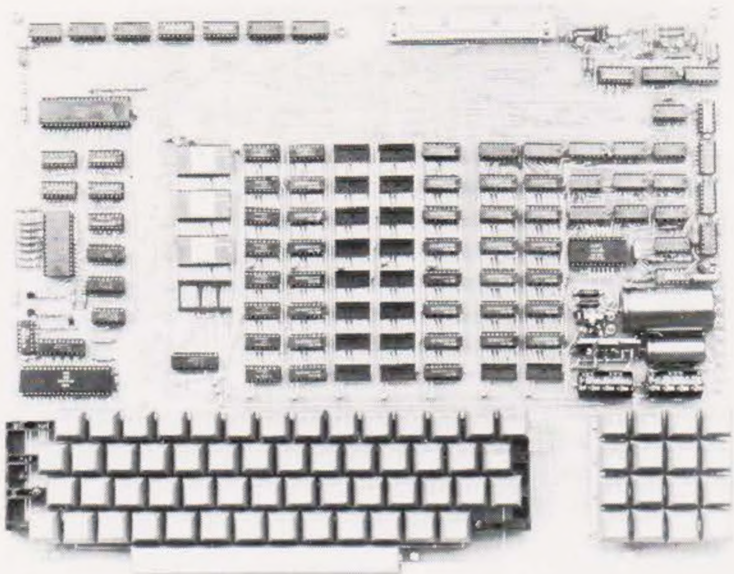
BATTLESHIPS!
Bring Out The Big Guns

Turing Machines

Number Cruncher

Triton Typecast


COMP 80
POWERTRAN
COMP 80



computing today

VOL.1, NO.9
NOV 1979

Editor: Ron Harris B.Sc
Ed. Assistant: Henry Budgett
Art Director: Diego Rincon
Production: Dee Camilleri, Loraine Radmore, Paul Edwards, Tony Strakas, Joanne Barseghian.
Ad. Manager: Chris Surgenor
Ad. Representative: David Sinfield
Editorial Director: Halvor Moorshead

Distributed by Argus Distribution Ltd. Printed by LSG. Limited, Lincoln.

EDITORIAL AND ADVERTISEMENT OFFICE
145 Charing Cross Road, London WC2H 0EE. Telephone 01-437 1002/3/4/5

NEWS

DUPED BY TAPE?

If you want to duplicate your software for your local club or friends a service is being offered by Simon Stable Promotions. Virtually any taping standard can be accomodated and quantities of ten upwards cost 33p each. A single sample can be done for 50p. Library boxes are available at 10p each but these will increase the post charges. Please note that this service should NOT be used to rip off others peoples software as you have to sign an indemnity form with each order that leaves you liable for heavy penalties if caught.For more details please contact Simon Stable at 46 West End,Launton, Oxon or ring 08692–2831.

MILLION BIT BOX

Rostronics have been appointed UK distributors for Micromation and the first product being sold is the Megabox. This is a double density, dual disk system with either single or double density units. Compatible with all 8080 and Z80 S100 systems at up to 4MHz it uses the CP/M operating system. Format is to the IBM 3740 standard but a special model is available for the TRS 80. The cost for the 1Mb unit is £1350 and the 2Mb unit is available at £1750. The special TRS-80 version includes 32K of RAM and an RS232 port so it can stand alone with the processor unit. Further details are available from Rostronics at 118 Wandsworth High St, London SW18.

2020 PRICE CRASH

ITT have announced that the price of their 2020 version of the Apple is being cut in price. The new suggested retail prices are £867 for 16K, £931 for 32 K and £995 for 48K. These reductions have been caused by increased demand and cost saving through in-house manufacture of IC's. Expected soon is a larger disk for the small business and educational end of the market.

NEW DEALER GIVES NEWSLETTER

A and G Knight of Aberdeen, a television rental company who are moving into the micro business are offering a free newsletter. Among their product range are Nascom and Xitex systems.After hearing many complaints about various newsletters they decided to produce their own. As well as covering their product range the publication includes detail on connecting a Nascom to replace a radio teletype, an article on bubble memories and more besides. For information on both the newsletter and theie range contact them at 108 Rosemount Place,Aberdeen AB2 4YW.

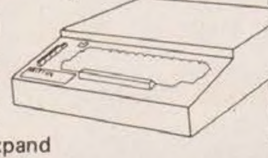
NEWBEAR BLAST OFF

For those of you with a desire to wipe out your own PROMS ready for new data Newbear are now stocking a range of UV lamps. These include a timer on all but the cheapest model and are of American origin. The cost for a six unit eraser is £56 going up to a 144 chip model at £1227. A leaflet is available from Newbear at 40 Bartholomew Street, Newbury, Berkshire.

TRANSAM COMPUTER PRODUCTS

TRITON SINGLE BOARD PERSONAL COMPUTER

Three new exciting expandable systems designed for ease of construction and flexibility.



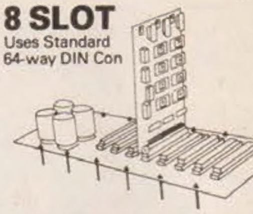
Kits come complete with case, power supply, full keyboard, PCB. All components available separately, see catalogue. Full hardware. Programming manual available. The system is easy to expand and is well supported. Features — 2, 2.5 or 7K Basic in Eprom (see catalogue).

- Single Board Holds up to 8K Memory
- Cassette Interface
- Three Firmware options
- Basic in Eprom
- Plug-in Expansion Boards

Personal Computer From **£286** + VAT

EXPANSION MOTHERBOARD

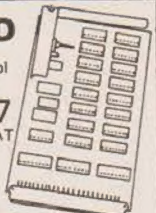
8 SLOT Uses Standard 64-way DIN Con



TRITON. Expand your Triton simply and easily with our new 8-slot motherboard complete with its own P.S.U. takes 8 plug-in Euro cards. Plug-in 8k RAM card and Eprom cards now available. Kit complete with PSU + 1 set connectors.

£50 + VAT

8K RAM CARD



Triton 8k Static RAM card kit uses 2114L low power 4k static RAMS. On board regulation. Mem jump select.

Compl Kit **£97** + VAT

PCB only £15 — RAMS £5.50
Kit less RAMS £35 incl 5SKTS & components

8K EPROM CARD

Triton 8k Eprom card kit designed to take up to 8 x 2708 Eproms (1k x8) as RAM card

PCB only £15
Kit less Eproms £31 **£97**
Eproms (blank) £9 Complete kit + VAT

BI DIRECTIONAL MATRIX PRINTER **£595** + VAT

The BD80 is a low cost, 80 column line printer with microprocessor control to provide excellent reliability and performance

- 5 x 7 Matrix
- 10 Char per inch
- 6 lines/inch
- 400 Char Buffer
- Full ASC II Char Set
- 10 lines/sec Paper Advance
- 112 Char/sec
- 84 lines per minute
- Self Test
- Fully Cased

A UNIQUE PRINTER FAST AND RELIABLE

Switch selectable baud rate from 110 to 9600 on a standard V24 and RS232 Interface — Send SAE for further details. Ideal printer for Triton or any system requiring high speed, reliable, hard copy. We can supply consumables.

S100 BOARDS

8k Static RAM board (450ns)	£123.75
8k Static RAM board (250ns)	£146.25
Z80 cpu board (2MHz)	£131.25
Z80 cpu board (4MHz)	£153.75
2708/27 16 EPROM board	£63.75
Prototype boards and bare boards.	
Video display board (64 x 16, 128 u/l Ascii).	£108.75
Disk controller board	£131.25
K2 disk operating system 8" disk	£56.25
Assemble/z Macro Assm 8" disk	£37.50

VISIT US FOR DEMONSTRATION

ITHACA Pascal/Z

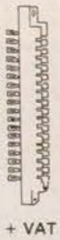
build your own Pascal Micro Development system. IEEE-SIDO bus system using DPSI mainframe. Supports K2, assemble/z and pascal/z on 8" disk.

your London dealer!

We stock the full range of ITHACA products.

PCB CONNECTORS

Edge connectors gold contact double sided PCB connectors



.1"	Price	.156"	Price
22/44	£3.20	6/12	£1.25
25/50	£3.60	10/20	£1.50
28/56	£3.90	12/24 (Pet)	£2.00
30/60	£4.15	15/30	£2.20
85/70	£4.60	18/36	£2.30
36/72	£4.75	28/56	£2.65
40/80	£5.00	36/72	£3.30
43/86	£5.50	43/82	£3.90
50/100 (S100)	£5.80		£4.60

+ VAT

TRAP!

Triton resident assembly language package.

Links via the L6.1 monitor and new scientific basic to make Triton a stand alone development system. Trap is an 8k package in EPROM and resides on our EPROM card. Set of 8 x 2708 only £80 including documentation.

- EDITOR
- ASSEMBLER
- DISASSEMBLER
- SYMBOL TABLE
- CREATE
- BREAKPOINT
- SINGLE STEP
- TRACE
- PROGRAMME LOAD
- MONITOR

See catalogue for further details

COMPONENTS 74LXX

								SN74LS324N	1.80
SN74LS00N	.18	SN74LS54N	.21	SN74LS136N	.40	SN74LS194AN	1.89	SN74LS325N	2.55
SN74LS01N	.18	SN74LS55N	.21	SN74LS138N	.75	SN74LS196AN	.85	SN74LS326N	2.55
SN74LS02N	.20	SN74LS55N	.21	SN74LS139N	.75	SN74LS196N	1.20	SN74LS327N	2.55
SN74LS03N	.18	SN74LS63N	1.50	SN74LS145N	1.20	SN74LS197N	1.20	SN74LS352N	1.35
SN74LS04N	.20	SN74LS73N	.35	SN74LS148N	1.75	SN74LS221N	1.25	SN74LS353N	1.50
SN74LS05N	.26	SN74LS73N	.40	SN74LS151N	.85	SN74LS240N	2.20	SN74LS365N	.65
SN74LS08N	.20	SN74LS75N	.46	SN74LS153N	.60	SN74LS241N	1.90	SN74LS366N	.65
SN74LS09N	.22	SN74LS76N	.35	SN74LS154N	1.60	SN74LS242N	1.90	SN74LS367N	.65
SN74LS10N	.18	SN74LS78N	.35	SN74LS156N	1.25	SN74LS243N	1.95	SN74LS368N	.65
SN74LS11N	.26	SN74LS83AN	1.15	SN74LS156N	1.25	SN74LS244N	2.10	SN74LS373N	1.75
SN74LS12N	.25	SN74LS85N	1.10	SN74LS157N	.60	SN74LS245N	2.60	SN74LS374N	1.70
SN74LS13N	.55	SN74LS86N	.40	SN74LS158N	.99	SN74LS247N	1.25	SN74LS375N	.72
SN74LS14N	.89	SN74LS90N	.65	SN74LS160N	1.15	SN74LS248N	1.95	SN74LS377N	1.75
SN74LS15N	.25	SN74LS91N	.99	SN74LS161N	1.15	SN74LS249N	1.30	SN74LS378N	1.32
SN74LS20N	.20	SN74LS92N	.90	SN74LS162N	1.15	SN74LS251N	1.45	SN74LS379N	1.40
SN74LS21N	.26	SN74LS93BN	.65	SN74LS163N	.90	SN74LS253N	1.25	SN74LS381N	3.65
SN74LS22N	.26	SN74LS95AN	1.20	SN74LS164N	1.50	SN74LS257N	1.40	SN74LS386N	.57
SN74LS26N	.29	SN74LS96N	1.75	SN74LS165N	1.70	SN74LS258N	.95	SN74LS390N	1.98
SN74LS27N	.35	SN74LS107N	.39	SN74LS166N	1.75	SN74LS259N	1.45	SN74LS393N	1.50
SN74LS28N	.35	SN74LS109N	.39	SN74LS168N	1.95	SN74LS260N	.39	SN74LS395N	1.80
SN74LS30N	.25	SN74LS112N	.39	SN74LS169N	1.95	SN74LS261N	3.50	SN74LS396N	1.70
SN74LS32N	.27	SN74LS113N	.44	SN74LS170N	2.50	SN74LS266N	.39	SN74LS398N	2.75
SN74LS33N	.39	SN74LS114N	.44	SN74LS173N	2.20	SN74LS273N	1.85	SN74LS399N	1.60
SN74LS37N	.29	SN74LS122N	.79	SN74LS174N	1.15	SN74LS279N	.79	SN74LS424N	4.50
SN74LS38N	.29	SN74LS123N	.90	SN74LS175N	1.05	SN74LS280N	1.75	SN74LS445N	1.25
SN74LS40N	.25	SN74LS124N	1.50	SN74LS181N	2.75	SN74LS283N	1.80	SN74LS447N	1.25
SN74LS42N	.79	SN74LS125N	.65	SN74LS190N	1.75	SN74LS290N	1.80	SN74LS490N	1.95
SN74LS47N	.95	SN74LS126N	.65	SN74LS191N	1.75	SN74LS293N	1.80	SN74LS668N	.95
SN74LS48N	.95	SN74LS132N	.75	SN74LS192N	1.45	SN74LS295AN	2.20	SN74LS669N	.95
SN74LS49N	1.09	SN74LS133N	.39	SN74LS193N	1.75	SN74LS298N	2.20	SN74LS670N	2.70

MEMORY AND SUPPORT CHIPS — NEW LOW PRICES

(prices exclude VAT)

SUPPORT	
8212	2.20
8216	2.20
8224	2.80
8226	2.20
8228	4.20
8238	4.20
8245	11.00
8246	11.00
8251	5.00
8253	11.00
8255	5.00
8257	11.00
8259	12.50
8292	18.00
6820P	4.50
6821P	4.50
6850P	4.60
6852P	5.50
AY-5-2376	11.50
MC14411	12.00
M57109	12.43
M57160	10.00
M57161	10.00
TMS6011	5.00
81LS95	1.80
81LS96	1.80
81LS97	1.80
81LS98	1.80

RAMS	
2101	2.32
2102L-4	1.20
2111	2.32
2112	2.46
6810	4.08
8154	8.18
2114	5.50
2102L-3	1.60
74C920	11.00
74C921	11.00
74C929	11.00
4027	11.00
4044	14.70
4045	9.15
4060	7.00
2107	7.80
4116	8.00
4118	20.00
80P20	10.00
Z80CTC	10.00
Z80AP10	14.00
Z80ACTC	14.00

EPROMS	
1702	6.00
5204	8.00
2708	9.00
2516	28.00
2716	22.00
ROMS	
74S287	3.70
74S472	12.00
74S70	8.00
74S473	12.48
74S474	12.48
I/O	
2513	7.50
96364	10.95
14412	12.90
LINEARS	
LM301AH	.39
LM301AN-8	
LM (Mini Dip)	.30
LM308N	.99
LM309K	
LMT03	1.45
LM311H	1.29
LM318H	2.25
LM323K	6.00
LM324N	.79
LM339N	.54
LM555N	.30
LM556N	.75
LM709CN	.37
LM723CH	.58
LM723CN	.43
LM733CN	1.30
LM739CN	1.30
LM741CN-14	.33
LM741CN-8	.25
LM747CN-14	.79
LM747CH	1.19
LM748CN-8	.45
LM748CH	.46
LM1458H	.72
LM1458N-8	.48
LM1488D	.855
LM1489D	.85
LM1489AD	1.25
LM1496N-14	.99
LM3302N	.65
LM3401N	.65
LM3403N	1.20
LM3900N	.54
TL080CP	1.49
TL081CP	.69
TL082CP	1.29
TL083CN	1.65
TL085CN	1.69

VOLT REGS	
7805	.90
7812	.90
7815	.90
7824	.90
7805K	1.50
7812K	1.50
7815K	1.50
7824K	1.50
7905	1.10
7912	1.10
7915	1.10
7924	1.10
7905K	1.80
7912K	1.80
7915K	1.50
7924K	1.80
DIL SKTS	
8DIL	.14
14DIL	.15
16DIL	.17
18DIL	.24
20DIL	.27
24DIL	.30
28DIL	.36
48DIL	.50

CRYSTALS	
100K	3.00
200K	3.70
1MHZ	3.80
1008K	3.50
1843K	3.00
2MHZ	1.50
2457K	3.05
3276K	2.70
3MHZ	3.05
4MHZ	2.10
4.43M	1.00
5MHZ	2.70
6MHZ	2.70
7MHZ	2.70
7.168M	2.50
8MHZ	2.70
10MHZ	2.70
10.7M	2.70
18M	2.90
48M	2.90
CMOS	
CD4011	.15
CD4040	.79
+ full range	

MISCLE	
2513	7.50
TMS6011	5.00
MC14411	12.00
MC14412	12.90
96364	10.95
CPU's	
8080	6.33
6800	10.00
Z80	8.00
Z80A	15.00
8085	12.95
6502	8.00
SCMP II	10.00
6802	13.95
9900	30.00
W/WRAP SKTS	
8DIL	.20
14DIL	.35
16DIL	.42
18DIL	.60
24DIL	.52
28DIL	.74
40DIL	.96

16 x 214L only £79
8 x 4116 only £58

TRITON DOCUMENTATION

available separately as follows, prices include p & p

Triton manual — detailed circuit description and constructional details plus user documentation on level 4.1 monitor and basic	£5.70
L4.1 listing — listing of 1K monitor and 2K tiny basic	£4.20
L5.1 user documentation on level 5.1 firmware	£1.20
L5.1 listing — listing of 1.5K monitor and 2.5K basic	£5.20
L6.1 user documentation on 7K basic interpreter	£1.80
Motherboard, 8K RAM and 8K EPROM constructional details	SAE

User group newsletter subscription £4 per annum Triton software — Send SAE for list of programs available for Triton.

HOME COMPUTING CATALOGUE

If you're in town, visit our showroom in Chapel Street, next to Edware Road tube station. We have Tritons on display plus a comprehensive range of components and accessories, specifically for personal computer users. Books, mags, tapes, data, cables plus much more. Showroom open 6 days a week. (Half day Thurs. from 1.30).

NEW
A4 size catalogue filled with our latest products
40p + SAE
ALL PRICES EXCLUDE VAT

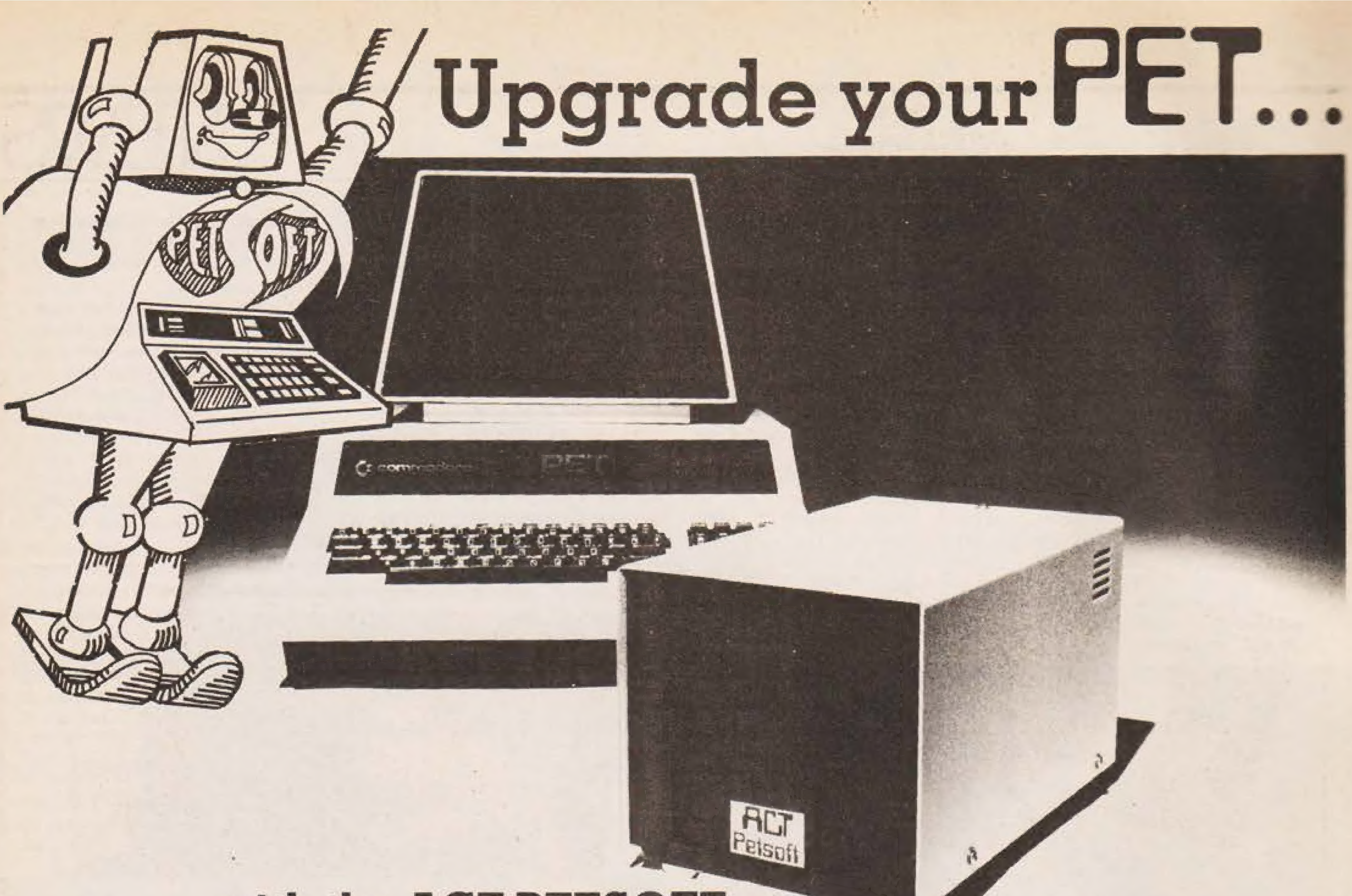
TRANSAM

VAT ALL PRICES EXCLUDE VAT

TRANSAM COMPONENTS LTD.
12 CHAPEL STREET
LONDON, NW1 TEL: 402 8137

ALIEN—A REVIEW

What, you may ask is a film review doing in a computer magazine! Well the answer is quite simple.When we published the Dateline 5000 program a couple of months ago we wanted to get hold of a few pictures of spacecraft to illustrate the article with, so we tried all the companies who were making sf movies. We didn't get any pics but we did get an invite to the Press show of Alien. By now you have probably heard about the horror aspect of the film in a variety of reviews but this aspect aside it must surely rate as one of the best films of the year, or even the decade.The whole film is set on a ship that has to make a landing to check out a distress call. It could be in the 18th century but instead it is in the future, and in space, not that this detracts in any way from the story. The central plot is the classic 'who gets got next' story and the 'getter' is of course the Alien. The superb use of the special effects in the film add to the story rather than turning it into yet another fantasy type film of the Star Wars or Gallactica type. Yes, this rates as a true sf movie and as a necessary part of the story there is a quantity of gore. The novel, which I had read, does not in any way spoil the horror element of the film,it just fills in some of the details which have to be left out because of time, I wonder just what was in the 13 minutes that they cut? The second very good reason to go and see the film is the prescence of a young lady called Sigourney Weaver, making her film debut with a stunning performance as Ripley. It is mildly ironic that out of the crew of seven on the ship (plus a cat) she is one of the survivors. Alien is undoubtedly a true piece of science fiction, possibly an epic and if you don't mind being scared half to death you really should go and see it. Alien, cert X runs for 1 hour and 56 minutes.



Upgrade your PET...

... with the ACT PETSOFT PROFESSIONAL DISK SYSTEM

A complete stand-alone system for the Commodore PET allowing up to 800,000 bytes of mass storage online. Designed by Compu/Think for business use, this powerful double density system offers complete random or sequential file access and support.

The Disk Operating System is in a ROM which plugs directly into 16K and 32K (new ROM) PETs, or via an Expandamem memory expansion board for 8K PETs. The Disk Operating System adds 16 extra easy-to-use commands to PETs BASIC. The Disk unit comes with a complete set of utility programs and a comprehensive manual.

- Supported by PETACT Business Software: Sales Accounting, Invoicing, Purchase Accounting, and soon Stock Control and Payroll
- More memory power for your money
- Reliable and easy to use
- Languages supported include BASIC, 6502 Assembler, FORTH, FIFTH, PLM, PILOT, CESIL and soon FORTRAN and PASCAL
- Wide range of PETSOFT programs including Payroll **£50** + Stock Control **£25**
- Supports the Pagemate Database **£299**

Prices exclude VAT

up to 800K online!

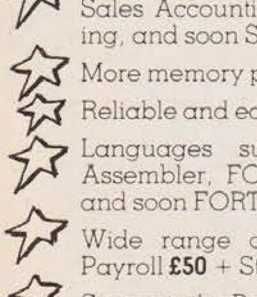
800K	Dual headed double density dual disk for 16K and 32K (new ROM) PETs	**£995**
400K	Double density dual disk for 16K and 32K (new ROM) PETs	**£840**
400K	Double density dual disk for 8K (old ROM) PETs. Requires Expandamem	**£795**
24K	Expandamem internal expansion memory board	**£320**

Prices exclude VAT.
PET is the trademark of Commodore who recommend PETACT Business Systems.

Try the ACT PETSOFT Professional Disk System and software at your PET dealer or write to us for full details and the name of your nearest stockist.

ACT Petsoft

Radclyffe House, 66-68 Hagley Road, Edgbaston, Birmingham, B16 8PF. Telephone: 021-455 8686 Telex: 339396



Please send me details of the PETSOFT PROFESSIONAL DISK SYSTEM

My name is ....

I live at ....

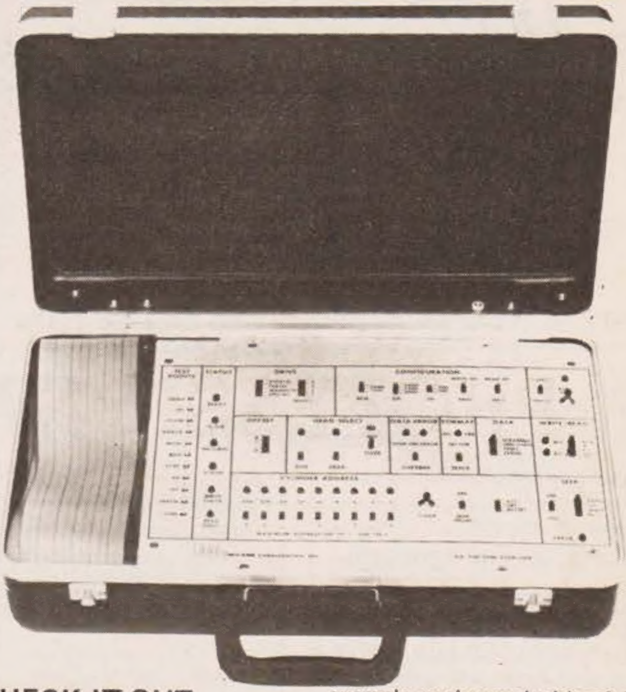
.... Postcode ....

I have a new/old ROM PET

NEWS

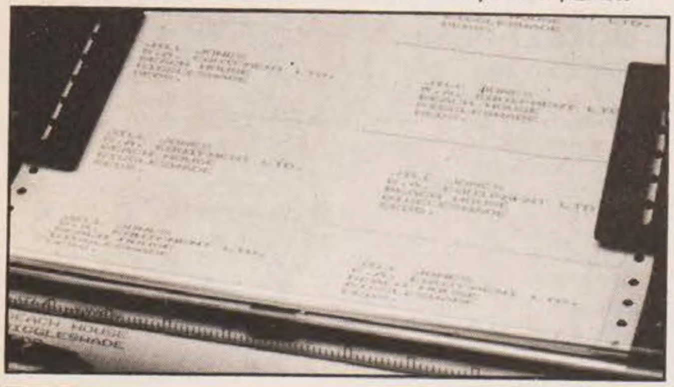
GET IT WITH APPLE TOO!

Pacsal is now available. Microsense, the master UK distributor for the Apple are now about to start distribution of Pascal (starting September). The language is an implementation of the standard and includes extensions for graphic and filing as well as a text editor. The language requires 48K of RAM,at least one disk and a full VDU if the 80 character line is required. The standard TV only gives the left or right 40. The new language is supported by a system card that adds 16K to the RAM and allows a choice of languages including Pascal to be used. Further features include an auto-start facility for either BASIC or Pascal. The price is a mere £296 and you can use old software under the new system. For more details please contact Microsense at Maxted Road, Maylands Avenue,Hemel Hempstead, Herts.



LABEL YOUR OWN!

A range of self-adhesive labels has just been launched by the firm of Fisher Clark. Made in a new material that overcomes the problems of tractor fed printers they are available in a wide range of sizes and numbers per sheet. Special sizes and shapes are available to order and a nationwide stockist network has been set up for their distribution. For details please contact Fisher Clark at 102 West Street, Boston, Lincs.

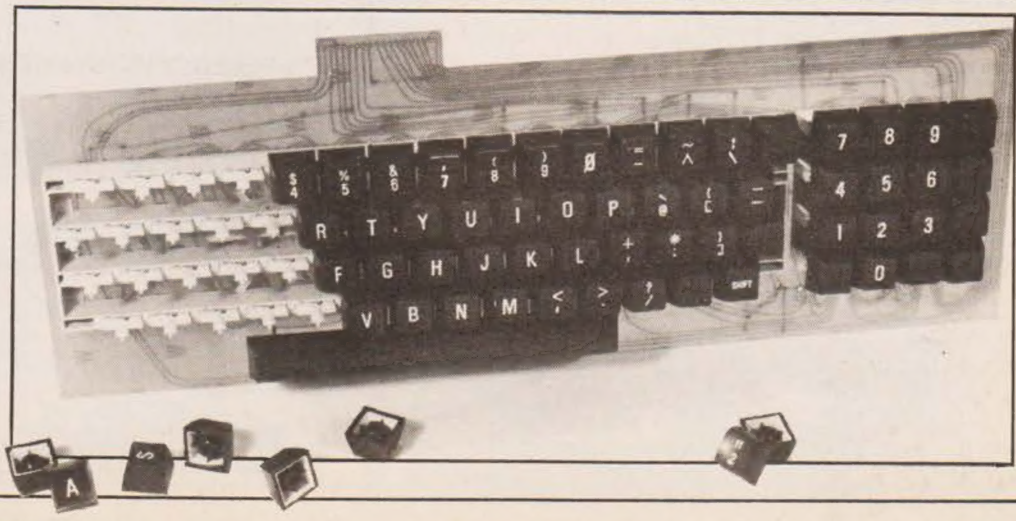


CHECK IT OUT

Got a problem with your disk? Worry no longer because a new range of peripheral test gear has been introduced.Made by Wilsons and distributed by Rack Data the units cost between £700 and £1300 and check out all functions of disk tape and a variety of other mass storage devices. The floppy disk exerciser for example can be used with mini or standard drives and is user programmable. It displays the current status at all times and provides three test patterns for data checking. All the units in the range are portable and self-powered, ideal for the field engineer.Details available from Rack Data at Rose Industrial Estate, Cores End Road, Bourne End, Bucks or ring 06285-27117.

KEY TO SUCCESS

A new range of modular keyboards is available from Devlin Electronics. By using a one piece moulding they have eliminated the need for metal frames. Both standard 60 and 16 way units are available in either flat or sloping types and they can be supplied to suit a users own PCB or on a PCB to suit PET,ASCII or XY matrix applications. All keytops are double shot engraved and prices range from £7.20 for the 16 way unit to £26 for the 60 way. Further details from Devlin at Caroline Court, Cranbourne Lane, Basingstoke Hampshire.



MICRO COURSES

A large number of micro based courses have been scheduled for the next few months. First we have a three day course being run by Microsystem consultants of PO Box 65, Camberly, Surrey. Based on the AIM 65 processor the course runs 5-7 November at the Euro Crest Hotel in Maidenhead. Cost is £185+VAT and includes lunch, refreshments and course material. You can ring Microsystem on Camberly 27417.

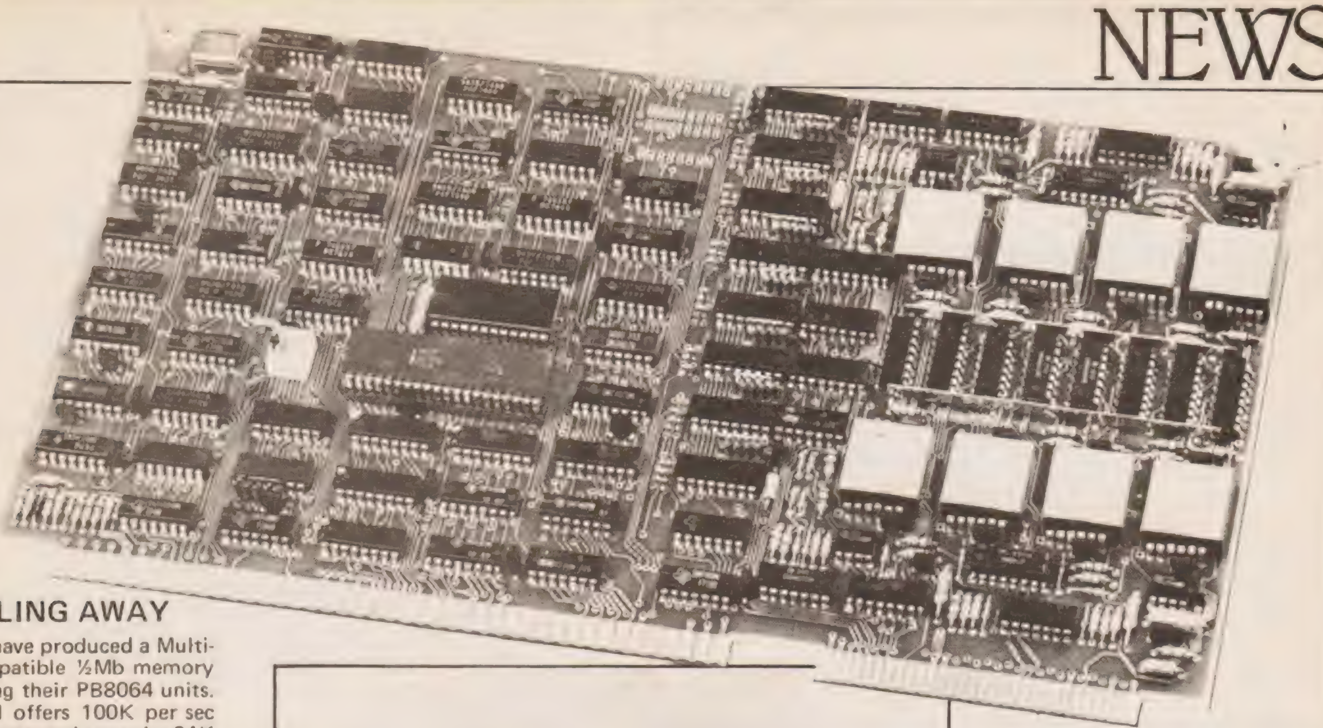
A one day course is being run at Thames Poly on 21 Nov. Called an Introduction To Microprocessors in Engineering it costs £32 and interested parties should contact Mr Tyler at the Poly in Wellington Street, Woolwich or ring him on 01-854 2030 extn 396.

Next on the agenda is a three day course from 5-7 November called Micr Computers, Application and Planning.This is being run by the Coventry and district Engineering Employers Association.The fee is £138 incl of VAT and lunches and the course is based around the Commodore PET.Anyone interested should contact Mr Jackson at Woodland Grange, Leamington Spa CV32 6RN or ring 0926–36621.

Finally a bumper bundle of courses from the Central London Poly and ICS.Included are courses on Pascal,Computer Graphics,Trouble Shooting and a whole lot more. Details are available from ICSP at Pebblecoombe, Tadworth, Surrey KT20 7PA or ring on Leatherhead 79211.

PETSOFT GOES HARD

Petsoft are now moving into the hardware market with an exclusive UK distributorship of the Compu/Think disks.The first product is a 400K double density,dual drive floppy. At around £795 it is faster and cheaper than other commercially available units. Also being sold are the Expandamem boards for PET,KIM and AIM systems. Another recently announced product from Petsoft is a cassette based magazine called CURSOR. Each tape contains at least 5 new programs and has a full graphics cover and contents table. Full documentation is supplied with each tape and also included is news on all PET related products. The cost of a years sub to the magazine is £36 for ten issues, a trial sample is available for £4. For details contact Petsoft at PO Box 9 Newbury, Berks.



BUBBLING AWAY

Plessey have produced a Multibus compatible ½Mb memory card using their PB8064 units. The card offers 100K per sec transfer rate and uses the 64K device. A MIL spec version,the PBM90M offers 8Mb in an ATR module as an alternative. Prices range from £500 to £1300. Also in the bubble field this month are GR Electronics of Newport who have produced a serially interfaced unit designated GR7000. The units are packaged with a PSU in 'black box' style and offer 10 to 40K with daisy chain capability.An integral V24/ RS232 connector offers Baud rates of up to 9600 giving a 30mS access time to any new start address. The need for special software is reduced by having integral firmware.Prices range from £750 for 10K to £1000 for 40K.

PET MEETS THE WORLD

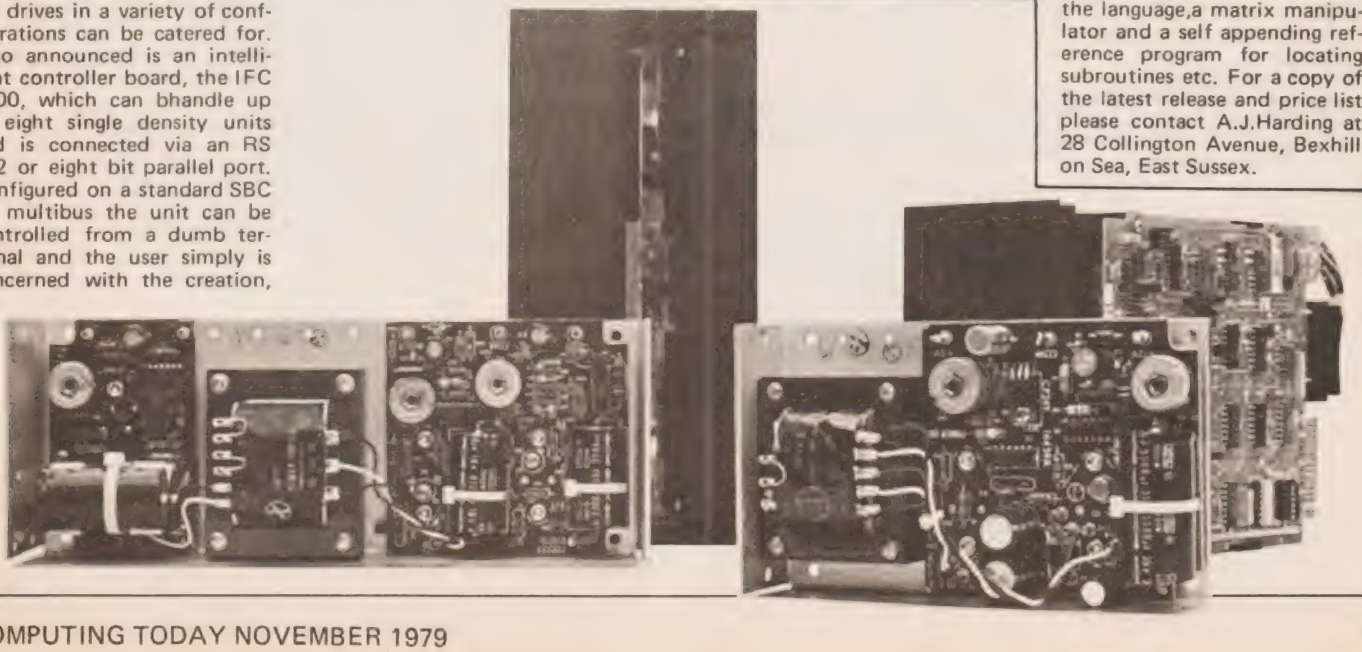
A communications unit is now available for the PET. Plugging into the user port it provides eight channels for either input or output or mixed.The device is called the Communicator and will allow you to use the PET as an intelligent control device for such systems as a burgular alarm or central heating. Supplied ready built at £135 it comes with software details for some examples. Further information from Mektronic Consultants at Linden House, 116 Rectory Lane, Prestwich, Manchester.

POWER ON WITH POWER ONE

HAL Computers are stocking a range of power supplies for both standard and mini disk units. IBM, Shugart and PerSci drives in a variety of configurations can be catered for. Also announced is an intelligent controller board, the IFC 8400, which can bhandle up to eight single density units and is connected via an RS 232 or eight bit parallel port. Configured on a standard SBC 80 multibus the unit can be controlled from a dumb terminal and the user simply is concerned with the creation, appending and deletion of files as the operating system is built in. Details are available from HAL Computers at 133 Woodham Lane, New Haw, Weybridge, Surrey.

MORE FROM MOLIMERX

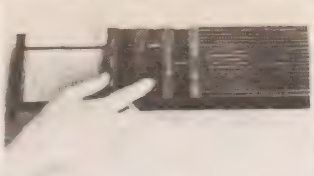
A.J.Harding have added yet more software to their already vast range for the TRS-80. Included in the new range is a software routine to allow you to pack 117740 bytes onto a standard diskette instead of the normal 89600 bytes. Also announced is a level 2 expansion adding 70 commands to the language,a matrix manipulator and a self appending reference program for locating subroutines etc. For a copy of the latest release and price list please contact A.J.Harding at 28 Collington Avenue, Bexhill on Sea, East Sussex.



24 TUNE DOOR CHIMES

DOOR TUNES £17.13 + VAT.

Waddington's Videomaster announce a doorbell than doesn't go Brrringgg, Ding Dong or Bzzzzz. Instead it plays 24 different classical and popular tunes. It will play the tune you select for your mood, the season or the visitor you are expecting to call. Door tunes is not only great fun and a wonderful ice breaker, but is also very functionally and beautifully designed to enhance your home. There is something for Christmas, something for your continental visitors or your relations from the states, and ever something for the Queen. Door tunes is easy to install and has separate controls for volume, tone and tempo.



T.V. GAMES

PROGRAMMABLE £29.50 + VAT.
COLOUR CARTRIDGE T.V. GAME.

The TV game can be compared to an audio cassette deck and is programmed to play a multitude of different games in COLOUR, using various plug-in cartridges. At long last a TV game is available which will keep pace with improving technology by allowing you to extend your library of games with the purchase of additional cartridges as new games are developed. Each cartridge contains up to ten different action games and the first cartridge containing ten sports games is included free with the console. Other cartridges are currently available to enable you to play such games as Grand Prix Motor Racing, Super Wipeout and Stunt Rider. Further cartridges are to be released later this year, including Tank Battle, Hunt the Sub and Target. The console comes complete with two removable joystick player controls to enable you to move in all four directions (up/down/right/left) and built into these joystick controls are ball serve and target fire buttons. Other features include several difficulty option switches, automatic on screen digital scoring and colour coding on scores and balls. Lifelike sounds are transmitted through the TV's speaker, simulating the actual game being played.
Manufactured by Waddington's Videomaster and guaranteed for one year.

EXTRA CARTRIDGES

ROAD RACE – £8.87 + VAT.
Grand Prix motor racing with gear changes, crash noises

SUPER WIPEOUT – £9.17 + VAT.
10 different games of blasting obstacles off the screen

STUNT RIDER – £12.16 + VAT.
Motorcycle speed trials, jumping obstacles, leaping various rows of up to 24 buses etc.

NON PROGRAMMABLE TV GAMES

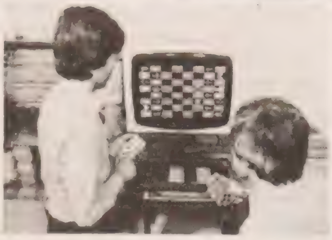
6 Game – COLOURSCORE II – £13.50 + VAT.

10 Game COLOUR SPORTSWORLD £22.50 + VAT.

CHESS COMPUTERS

STAR CHESS – £55.09 + VAT.
PLAY CHESS AGAINST YOUR PARTNER.

using your own TV to display the board and pieces. Star Chess is a new absorbing game for two players, which will interest and excite all ages. The unit plugs into the aerial socket of your TV set and displays the board and pieces in full colour (or black and white) on your TV screen. Based on the moves of chess. It adds even more excitement and interest to the game. For those who have never played Star Chess is a novel introduction to the classic game of chess. For the experienced chess player, there are whole new dimensions of unpredictability and chance added to the strategy of the game. Not only can pieces be taken in conventional chess type moves, but each piece can also exchange rocket fire with its opponents. The unit comes complete with a free 18V mains adaptor, full instructions and twelve months guarantee.

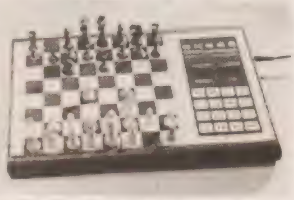


CHESS CHALLENGER 7 – £85.65 + VAT.
PLAY CHESS AGAINST THE COMPUTER.

The stylish, compact, portable console can be set to play at seven different levels of ability from beginner to expert including "Mate in two" and "Chess by mail". The computer will only make responses which obey international chess rules. Castling, en passant, and promoting a pawn are all included as part of the computer's programme. It is possible to enter any given problem from magazines or newspapers or alternatively establish your own board position and watch the computer react. The positions of all pieces can be verified by using the computor memory recall button.
Price includes unit with wood grained housing, and Staunton design chess pieces. Computer plays black or white and against itself and comes complete with a mains adaptor and 12 months guarantee.

OTHER CHESS COMPUTERS IN OUR RANGE INCLUDE

CHESS CHAMPION – 6 LEVELS £47.39 + VAT.
CHESS CHALLENGER – 10 LEVELS – £138.70 + VAT.
BORIS – MULTI-LEVEL TALKING DISPLAY £163.04 + VAT.



ELECTRONIC CHESS BOARD TUTOR £17.17 + VAT.

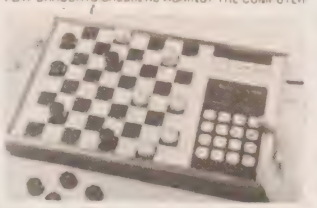
A special bulk purchase of these amazing chess teaching machines enables us to offer them at only £19.75 less than half recommended retail price. The electronic chess tutor is a simple battery operated machine that can actually teach anyone to play chess and improve their game right up to championship level. This machine is not only for total beginners but also for established players wanting to play better chess. Unit contains the electronic chessboard with 32 chess pieces, a 64 page explanatory booklet and a set of 32 progressive programme cards including 6 beginners cards, 16 check mate positions, 9 miniature games, 5 openings, 3 end games, 28 chess problems and 2 master games.

DRAUGHTS COMPUTERS

CHECKER CHALLENGER 2 LEVELS £43.00 + VAT.
4 LEVELS £78.00 + VAT.

The draughts computer enables you to sharpen your skills, improve your game, and play whenever you want. The computer incorporates a sophisticated, reliable decision-making microprocessor as its brain. Its high level of thinking ability enables it to respond with its best counter moves like a skilled human opponent. You can select offence or defence and change playing difficulty levels at any time. Positions can be verified by computer memory recall. Machine does not permit illegal moves and can solve set problems. Computer comes complete with instructions, mains adapator and twelve months guarantee.

PLAY DRAUGHTS/CHECKERS AGAINST THE COMPUTER



FOR FREE BROCHURES – SEND S.A.E

For FREE illustrated brochures and reviews on TV and chess games please send a stamped addressed envelope, and state which particular games you require information on.
Callers welcome at our shop in Welling – demonstrations daily – open from 9am-5.30pm Mon-Sat (9am-1pm Wed)
To order by telephone please quote your name, address and Access/Barclaycard number.
Postage and Packing FREE

AJD DIRECT SUPPLIES LIMITED, Dept **CT 11** 102 Bellegrove Road, Welling, Kent DA16 3QD. Tel: 01-303 9145 (Day) 01-850 8652 (Evenings)

EPROM ERASERS

NEW

EX-STOCK

MODEL UV141 EPROM ERASER

- Fast erase times (typically 20 minutes for 2708 EPROM).
- 14 EPROM capacity.
- Built-in 5 to 50 minute timer to cater for all EPROMS.
- Safety interlocked to prevent accidental exposure.
- Convenient slide-tray loading of devices.
- 'MAINS' and 'ERASE' indicators.
- Rugged construction.
- Priced at only £78 + VAT (price includes delivery).

MODEL UV140 EPROM ERASER

Similar to Model UV141 but without timer
Low price at only £61.50 + VAT (price includes delivery).

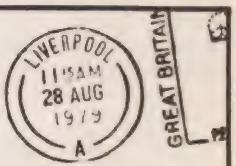
WRITE OR TELEPHONE FOR FURTHER INFORMATION OR SEND CHEQUES/OFFICIAL COMPANY ORDERS TO:

GP Industrial Electronics Limited

Skardon Works, Skardon Place, North Hill, Plymouth PL4 8HA. Telephone: Plymouth (0752) 28627

TRADE & EXPORT ENQUIRIES WELCOME

Microcomputer Mail Order



All your microcomputer requirements can be bought with confidence by mail order from MICRODIGITAL, one of the largest and longest established computer stores.

Most orders are despatched same day as receipt, if not a note explaining what the supply situation is. If we cannot supply within 30 days we will, on request, make an immediate cash refund.

Access and Barclaycard orders are welcome either in writing or over the phone. Your account will not be charged until the goods are despatched.

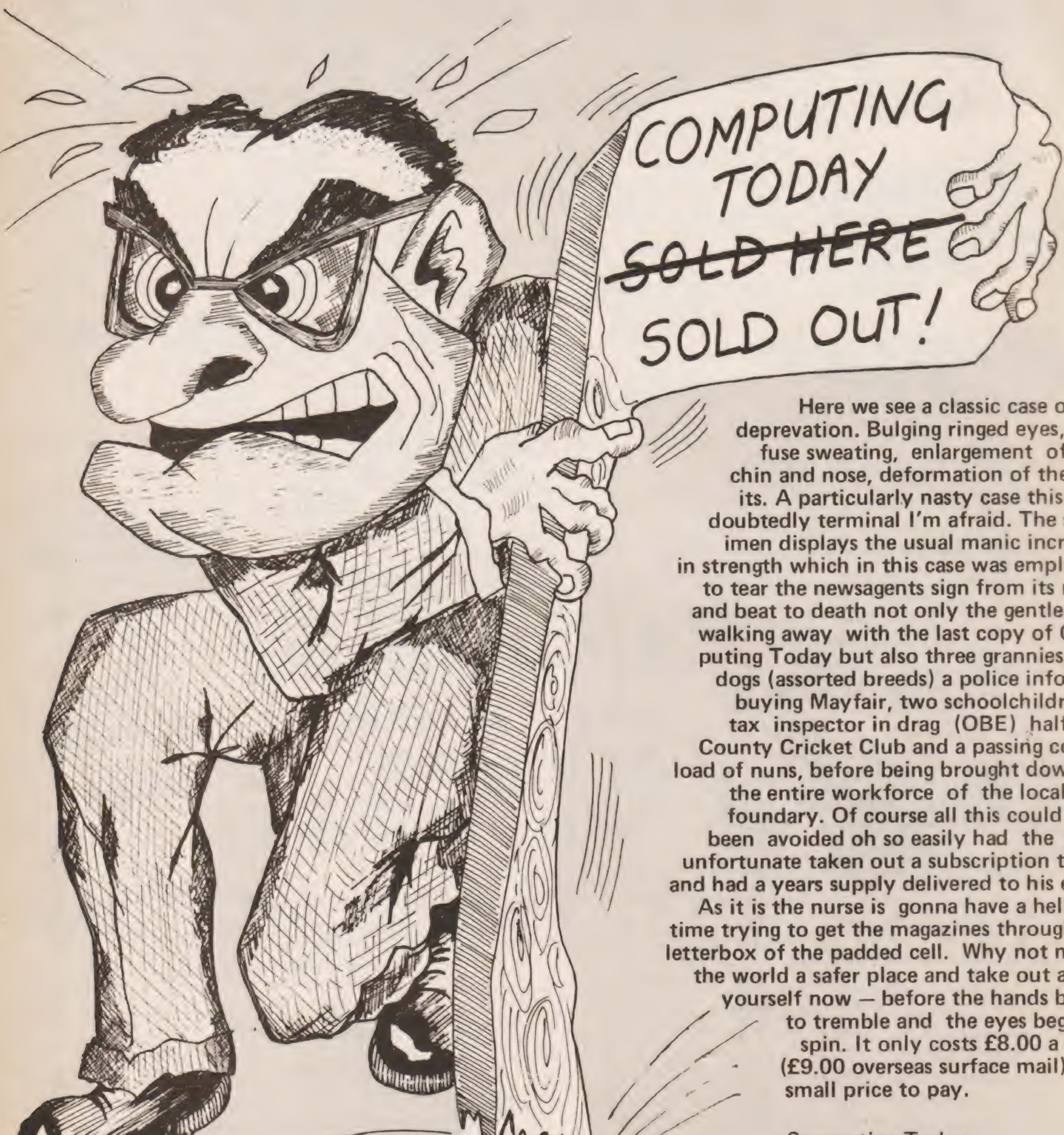
Official orders of over £10 are also welcome. With normal 30 days trade credit extended to bona fide commercial and government organisations.

If you do not have our brochures, write or phone today for free copies by return.

MICRODIGITAL

MICRODIGITAL LIMITED
FREEPOST (No stamp required) LIVERPOOL L2 2AB
TEL: 051-236 0707 (MAIL ORDER 24 HOURS A DAY)
TEL: 051-227 2535

ARRRGGGHHH.....



Here we see a classic case of CT deprevation. Bulging ringed eyes, profuse sweating, enlargement of the chin and nose, deformation of the digits. A particularly nasty case this. Undoubtedly terminal I'm afraid. The specimen displays the usual manic increase in strength which in this case was employed to tear the newsagents sign from its roots and beat to death not only the gentleman walking away with the last copy of Computing Today but also three grannies, five dogs (assorted breeds) a police informer buying Mayfair, two schoolchildren, a tax inspector in drag (OBE) half the County Cricket Club and a passing coach load of nuns, before being brought down by the entire workforce of the local iron foundary. Of course all this could have been avoided oh so easily had the poor unfortunate taken out a subscription to CT and had a years supply delivered to his door. As it is the nurse is gonna have a hell of a time trying to get the magazines through the letterbox of the padded cell. Why not make the world a safer place and take out a sub yourself now — before the hands begin to tremble and the eyes begin to spin. It only costs £8.00 a year (£9.00 overseas surface mail) — a small price to pay.

Computing Today
Subscriptions Dept.,

M.A.P. PO Box 35,
Bridge Street,
Hemel Hempstead,
Hertfordshire.

Giles Gummer

The art of Computability is the subject of this article. We also present a simulation program to try it out

One of the questions that Computer Scientists ponder in their baths is "What questions can computers never answer?". Much research has gone into describing simple "computers", because once such a machine has been described, the task of defining what problems it can (or can't) solve can proceed, much as algebraic proofs are built on the basic ideas of lines and angles.

Of course one could expect that the classes of unsolvable problems fo each machine would be entirely different, but it is one of the triumphs of the subject that this can be proven not to be the case. Although many different approaches to the task of defining a machine have been made, starting from abstract mathematics such as the theory of recursive functions, or from a simple, easily visualised physical model, like the Turing Machines I am going to describe, or from a mixture of both like the register machines of Marvin Minsky, each can be shown to be equivalent to the others. Problems that can be solved by one "machine" will be amenable for solution by another, and a problem that it is impossible for a Turing Machine to solve will also be unsolvable on a Minsky Machine.

This subject is known as the study of Computability, and a problem that can be solved by one of these universal computing systems is termed Computable.

Unsolvable Problems

Three questions that my pocket calculator cannot answer are:

1) What is the square root of −1?
2) What is 9 raised to the 9th power of 9?
3) Who won last year's Derby?

The first and last of these I will discount now, as they are merely beyond the scope of a machine that deals solely in real numbers. The second is "unsolvable" for rather a more subtle reason, since if I were given a calculator with a large enough display, I would be able to find the answer. A calculator is a finite machine (and so is an I.C.L. mainframe if it has no exchangeable storage media) so there are a finite number of 'states' it can exist in. Simply, if its internal circuitry consisted of n flip-flops, then it could assume no more than 2^n states, and its state at any one time could be given as an n-digit binary number.

Any finite machine can be fooled into giving the wrong answer merely by increasing the size or complexity of the problem it is posed. If my new large calculator was asked to find the factorial of nine to the (nine to the nine), it would fail where an even bigger one would not. Although I cannot comprehend the magnitude of the answer, nor guess the time that a calculator would take to produce it, nevertheless, (nine to the (nine to the nine))! = ? is a computable problem. All I need to solve it is integer arithmetic and very large internal registers and amounts of backing store.

If I now replace 'very large' with 'infinite', I have an infinite machine, with the knowledge of how to perform integer arithmetic (a program in firmware) and an unbounded amount of memory.

Turing Machines

Turing machines are also infinite. The invention (published 1936) of British mathematician Alan Turing, they consist of a program written as a sort of flow chart called a Finite State Diagram, which controls a single tape-head moving up and down a strip of magnetic tape that extends without bound in both directions. It is much easier to think about an infinite tape than to buy one, and this is one of the reasons why Turing Machines are still 'thought computers', and not built in hardware.

The tape-head moves over the tape, which remains stationary. The tape may be thought of as being divided into squares, each of which may contain a symbol. The head reads one symbol at a time, and moves one square at a time, either right or left. Fig 1 represents a tape and head, the zero's are blank tape. A Turing machine cycle consists of moving the tape head one square in the direction the program requires, reading the symbol now beneath the head, and perhaps overwriting it with another symbol. Data may be presented by writing it on the tape before setting the machine to work, and results appear as the symbols on the tape once the machine has halted at the end of its computation.

Finite State Diagrams

All the actions of a Turing Machine may be described (i.e. controlled) by means of a (Finite) State Diagram, though such diagrams are also used for other purposes. Fig 2 shows a simple example of a Diagram representing a program to find the parity of a binary number (whether it has an odd or even number of ones). It has two states, the large circles labelled E and 0. The machine starts in state E. The binary number is input to the machine one bit at a time. State E has two arrows labelled 0 and 1 pointing from it. These indicate the action to be taken upon reading in a 0 or a 1. If a 1 is read, the machine assumes state 0. If a 0 is read, it remains in state E. Similarly for state 0, a 1 takes the machine back to E, while a 0 leaves it where it is. The machine will count the number of 1's in its input, and exist in state E if that number was even, and in state 0 if it was odd.

This Diagram has written no symbols, nor has it tried to control a tape-head. Fig 3a shows a Diagram that does both of these, and which may be used to run a Turing Machine. When the Machine shown in Fig 3b starts under the control of Diagram 3a, the binary number written to the left of the A on the tape will be incremented twice.

It works like this. The machine starts in state L. The L indicates that it is to move left, and then look for a 0 or a 1. If it finds a 1 it is to replace it with a 0 (the symbol in the middle of the arrow labelled 1), and continue in the same state. Thus it will step left, replacing 1's with 0's until it encounters a 0. This it will replace with a 1, and the machine then changes state to R. The binary number has been incremented.

In the R state, the machine steps right, looking for an A or a B. It will ignore any 1's or 0's it reads, and will even-

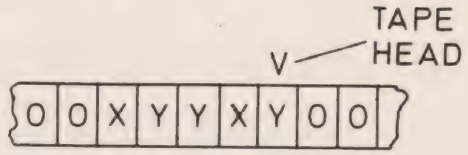


Fig.1. The tape on a Turing Machine extends to infinity, both left and right.

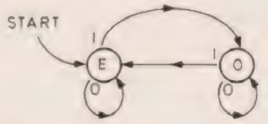


Fig.2. A simple example of a Finite State Diagram.

tually find the A marking the end of the number. The Diagram shows that the machine must replace this with a B and start again on the left search for a 0. The tape will now look like Fig 3c. The head will next move left, read the 0, replace it with a 1 and start on the right search. It immediately finds the B it wrote two steps before, and on reading this the machine halts (Fig 3d). Two has now been added to the number on the tape.

Programming Turing Machines

The second example above operated on only one number, a binary integer with the symbol A marking the least significant bit. The number could grow as large as it pleased, since the tape extended without bound to the left (and right). If two or more numbers are needed, they must be delimited in some way, and extra symbols may be brought into use as markers. Fig 4a shows one layout for three binary numbers. The X's mark the start of virgin tape.

Since we have introduced more symbols, there is no reason why the numbers on the tape need be represented in binary. Denary and Hexadecimal are just as legal, but in fact the base which is easiest to program in is Unary. In this base the number n is represented as n ones in a line, suitably delimited. It is a very primitive and simple counting system. Fig 4b shows the tape of 4a written in Unary. Fig 4c is a Diagram to increment a Unary number placed anywhere in a string of such numbers delimited as in Fig 4b. The tape head must start on the number or on the marker to its right. The diagram has seven states and four symbols; the same problem with numbers in binary would have been considerably more complicated. In general, you can write state diagrams with fewer states by employing more symbols, and vice-versa. The symbols used in a diagram and on a tape to solve a problem are called the alphabet of the Turing Machine that performs the solution. Taking Fig 3a and modifying it to add five to the number on the tape can be done by enlarging the alphabet (Fig 5a), or by adding more states and some data on the tape (Fig 5b).

Fig 5c shows the program and tape for adding two Unary numbers (very easy!) and Fig 5d is a program to multiply two Unary numbers by repeated addition, showing also the tape at the beginning and the end of the calculation. By making a few additions to Fig 5d you can get a program that will square a number, or raise it to an arbitary power, or produce its factorial. Hence a fairly simply Diagram could be produced to calculate in Unary, the value of (nine to the (nine to the nine))!.

Universal Turing Machines

Without doubt the most important discoveries about the behaviour of Turing Machines, and therefore the other computing systems, stems from the study of Universal Turing Machines, or Turing Machines that simulate the action of other Turing Machines. The problem is this: produce a state diagram for a Turing Machine which will be able to simulate in a recognisable way the action of any Turing

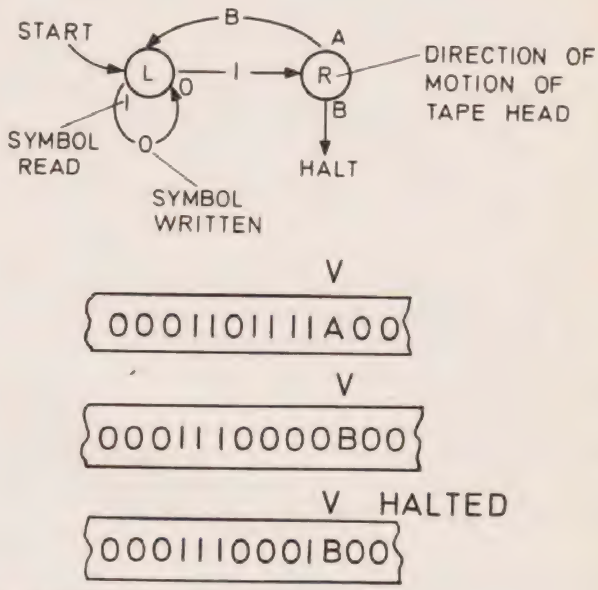


Fig.3. A Finite State Diagram that controls tape. The various stages are shown.

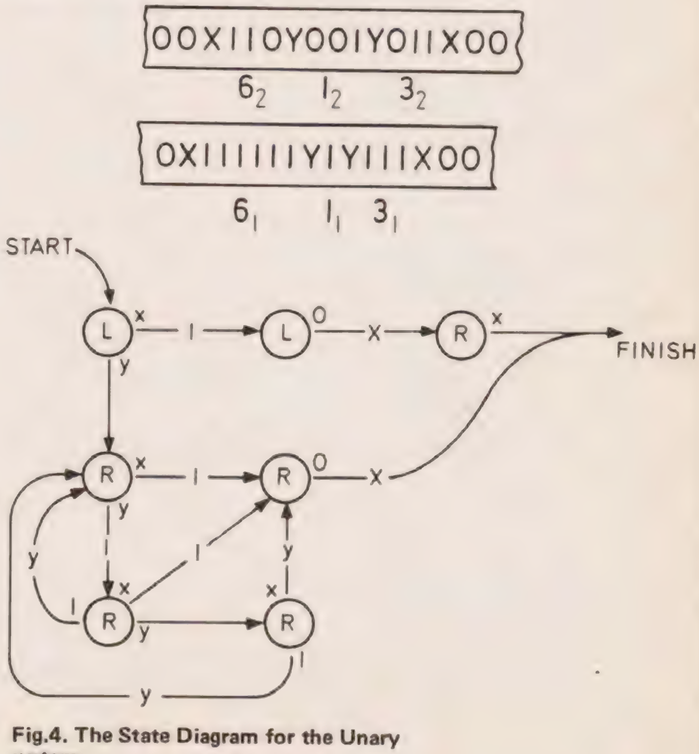


Fig.4. The State Diagram for the Unary system.

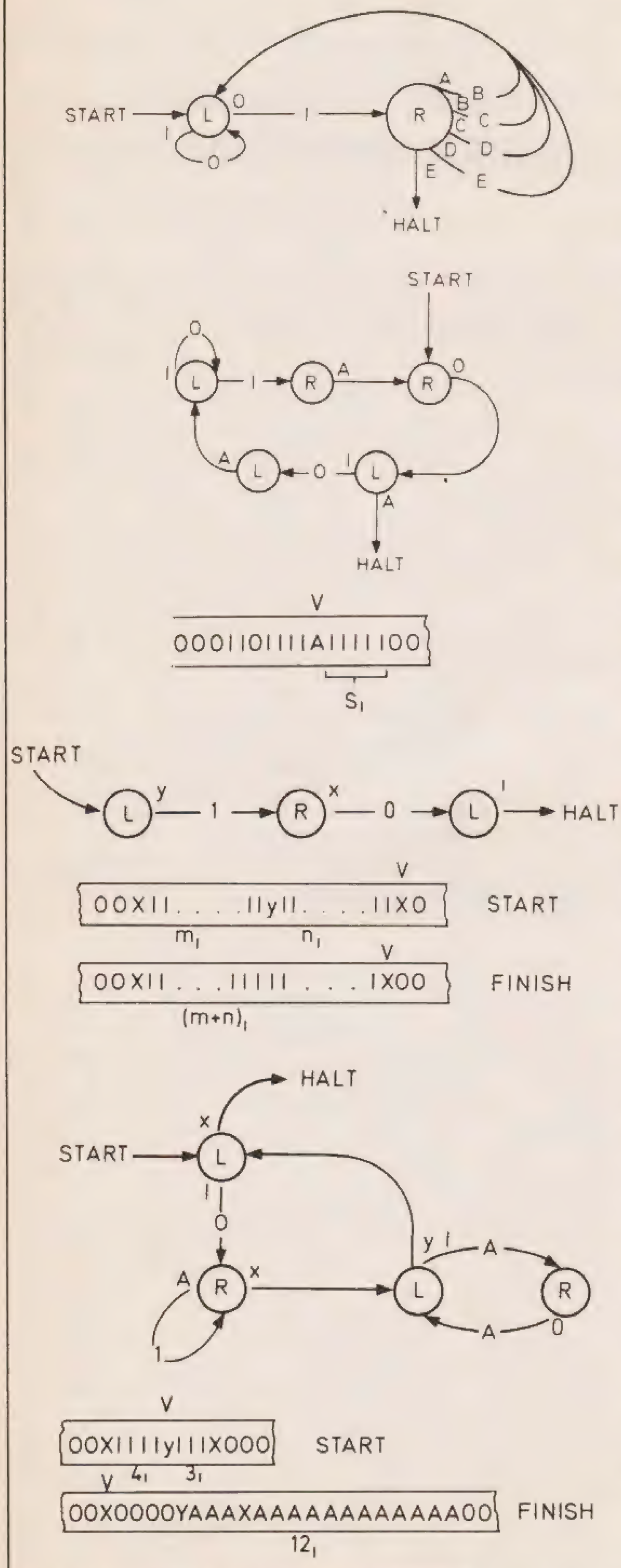


Fig.5. State Diagrams for handling Unary calculations.

Machine. It will need 1) the tape symbols that the simulated Turing Machine would start with and 2) a description of the state diagram that controls the simulated Machine, both written on the simulating Machine's tape as its data.

This is not too hard a problem. Once approach, that taken by M. Minsky in his excellent book 'Computation: Finite and Infinite Machines', Prentice Hall International Inc. 1972, is to keep a direct copy of the tape that the simulated Machine would produce. Certain simplifications can be made to the simulated Machine; its alphabet can be restricted to two symbols, since if its Diagram is written so as to consider these in groups of three (or four) then it can be made equivalent to a Machine with up to eight (or sixteen) symbols, and similarly it can be shown that a Turing Machine whose tape extends in only one direction can do anything that an ordinary Turing Machine can do. With these simplifications we need only simulate a Machine with the alphabet (0, 1), whose tape extends only to the left of a marker, and the space to the right of that marker is then available for describing the simulated Machine's State Diagram. Minsky keeps a 'File' of states, symbols recognised and corresponding actions, together with a record of the current state, and the current head position and input symbol. Different parts of the simulating machine then perform the actions of reading the input symbol, looking up the current state tofind what to do, if necessary changing the current state and writing an output symbol, and moving the simulated tape head.

Because some Turing Machine will be able to perform the solution of any given computable problem, this simulating machine will be able to perform the solution of any computable problem, given the correct data. It is thus called a Universal Turing Machine. Minsky manages to pack its Diagram into 23 states with an alphabet of 8 symbols, and if you think that is small, his book goes on to describe, amongst other things, a 7 state 4 symbol Universal Machine.

The Halting Problem

I started this article by asking "What problems can Computers never solve?", and the Halting problem is held to be one such. The stumbling-block remains that we cannot prove a Turing Machine to *be* a Computer, or a Computer to *be* a Turing Machine, in the same way that we cannot prove Newton's laws of motion. Nevertheless, the evidence in favour of both Newton (with due consideration given to Relativity) and Turing is considerable; in our case particularly the astonishing equivalence of Turing Machines and the other universal systems that model computability.

The Halting Problem concerns the existence of a particular Turing Machine, call it A, which, when presented with a description of another Turing Machine M, and the data that M starts with, will determine whether or not M will ever halt when started on its tape. It is quite possible for a Turing Machine never to reach the halt state when started on some tape, and it would be of value to know before we set a machine going on a lengthy calculation whether or not it is going to give us an answer. The description of M given to A could be in exactly the same form as the Universal Machine described above would need to simulate M.

Now the machine A which will tell us whether the Machine its data describes will ever halt may take a long time to decide, and indeed it itself may never halt. So perhaps the first thing to do is to give A a description of itself, and ask it if it will ever halt when set going on a tape containing a description of M. We now have A studying a simulated version of A studying a simulated version of M. If A exists,

and works, it will at some point now halt, saying 'Yes, I will halt and answer your question about M', or 'No, I will never halt'.

Sadly at this point we can shoot ourselves down. By sabotaging our machine A in a legal fashion by attaching to the 'YES' halting exit an infinite loop, we have a new machine B which acts like A except that it can only halt if the machine it is studying does not halt, and it takes the 'NO' exit. If B is set to studying the behaviour of itself, we now have a logical impossibility. For B can only halt if the machine it is studying does not, but it is studying itself, and will thus halt if it does not halt, and fail to halt if it halts! These are the only possibilities, and both are contradictions. This means that B cannot exist, and since if A exists, we can easily construct B, then A cannot exist either, and the problem that A was to solve is not computable.

This problem is truly unsolvable, and by reducing other (equally abstract) questions about Turing Machines to problems whose solution requires a machine like A we can build up a large family of provably unsolvable problems. Many of these problems have specific instances which are solvable; I can easily tell whether my Turing Machine is going to halt, but I have no general procedure for determining this for an arbitary Diagram and an arbitary tape, and by this definition the problem is non-computable. By studying these questions we are extending our knowledge of what computers as we understand them are ultimately capable of.

Nascom I Turing Machine Simulator

This program provides an animated simulation of a programmable Turing Machine. It runs on an unmodified Nascom I under T2. Facilities are provided to create a representation of the State Diagram within the Nascom, to write the memory tape's initial data configuration, and to step the Turing Machine through its computation either manually or automatically at variable speed.

The program executes from 0C50 and has three phases: the State Diagram, the Memory Tape and the Simulation.

The State Diagram

This section of the program (mostly the subroutine ENLIST) builds a chained Data Structure in core starting from 0F50. Fig 6 shows its logical and physical structure. Since entering a long state diagram is tedious, the first question the computer asks is 'N.S.D.?' (=New state diagram?). Type Y for a new one; any other key will leave the existing Data Structure undisturbed. This means that existing Diagrams may be reused or saved on tape for reloading later.

The computer first wants to know how many states there are in your new Diagram. Enter this as a hexadecimal number, remembering that the HALT state must count as one too.

Next you must enter information about each state, one state at a time. The computer will prompt you in the upper left corner of the screen with five mnemonics indicating which piece of data it is expecting. The mnemonics are:

1) STID for the state identity
2) NRSY for the number of symbols recognised by the state
3) IPSY for the code of the input symbol
4) DNST for the identity of the destination state
5) OPSY for the code of the output symbol

STID

States are identified by a two digit hexadecimal number. The direction of head movement associated with the state is given by the most significant bit of the identity number. ID's greater than 7FH are LEFT SEARCHING states. The other seven bits must be distinct for each state (97H and 17H are the SAME state) but need not be in a continuous numbered sequence. The state the Turing machine starts in MUST be the first state to be input to the computer.

NRSY

This Hex number defines how many times the diagram building program cycles round the next three prompts. The HALTing state is distinguished by being the only state to recognise no symbols.

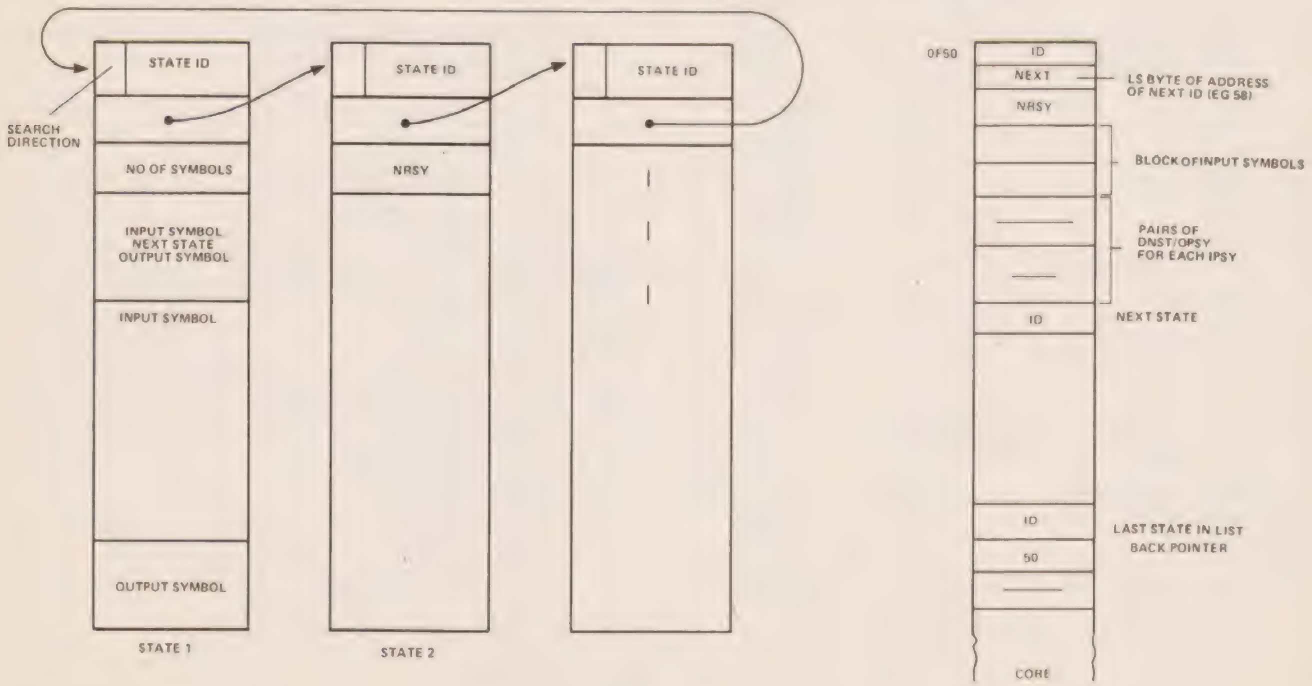
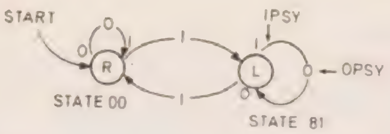
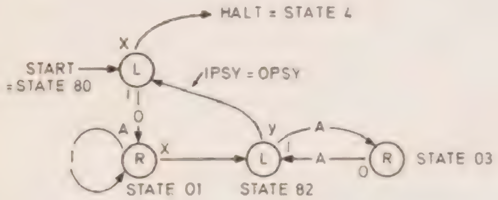


Fig.6. Finite State Diagram and data for the example program.



No. of States?	2	
STID	00	The start state; right searching
NRSY	02	or 2
IPSY	30	ASCII 0
DNST	0	i.e. this state
OPSY	30	=IPSY
IPSY	31	the second symbol recognised
DNST	1	the other state; 81 would have done as well
OPSY	31	
STID	81	the left searching state, no. 1
NRSY	2	
IPSY	31	order doesn't matter
DNST	1	
OPSY	30	
IPSY	30	
DNST	0	
OPSY	31	after this the machine will display the tape.

Start this one on a blank tape.



No. of States?	5 including the HALT state	
STID	80	start state, left searching
NRSY	2	
IPSY	58	ASCII X
DNST	4	4 will be the halt state
OPSY	58	
IPSY	31	
DNST	1	
OPSY	30	
STID	01	
NRSY	2	
IPSY	41	ASCII A
DNST	1	
OPSY	31	
IPSY	58	
DNST	2	
OPSY	58	
STID	82	state no. 2. left searching
NRSY	2	
IPSY	59	ASCII Y
DNST	0	
OPSY	59	
IPSY	31	
DNST	3	
OPSY	41	
STID	3	
NRSY	1	

Fig.7. State Diagrams and data lists for the program.

IPSY
The ASCII code of the symbol read in. During the simulation this is compared with the symbol currently under the tape head.

DNST
The identity of the state that the machine must assume upon recognising IPSY. This is again a two digit Hex number. Space has been left for implementing the simulation of NON-SEARCHING state diagrams, and the most significant bit of DNST is not used at present.

OPSY
The ASCII code of the symbol to be written under the tape head. If no change to the symbol is needed OPSY must be the same as IPSY, or strange characters will appear.

Errors
Errors in each line may be corrected by backspacing, but once Newline is pressed, the line is processed. If you make a mistake and cannot correct it for this reason, type FF in response to the next prompt. This will return you to the STID prompt and that whole state may be re-entered. If you type FF when the STID prompt is being displayed, the computer will step back to the previous state, so errors early in the procedure may be corrected. The error message 'NO!' is output and the monitor re-entered if you try to step back too many states. Fig 7 shows two State Diagrams and the corresponding input to the Nascom.

The Memory Tape
Fourteen lines of the Nascom's screen can be used to display the tape (which is of necessity not infinite. . .). After inputting the State Diagram the blank tape is displayed (filled with 0's) and can be manually written on to provide the initial tape configuration required. The following control characters are used:

- > step the tape head right once and remember the direction
- < step the tape head left once and remember the direction
- $ start the simulation with the tape and head as currently displayed

All other characters are written onto the tape, and the head is stepped on one square in the current direction.

The Simulation
The $ control character starts the simulation. At the end of each Machine cycle the program will look for characters from the keyboard to control the movement of the head. The following are recognised:

- M set to manual mode
- A set to automatic mode
- F speed up tape head
- S slow down tape head

Manual Mode
All keys except A will cause the Turing Machine to execute one machine cycle. A invokes automatic operation.

Automatic Mode

The computer will execute a delay loop between each Turing Machine cycle. The duration of this loop, and hence the speed of the tape head, can be altered by pressing S or F to slow it down and speed it up respectively. On long delays the key must be held down until the program executes the CALL which reads the keyboard or the computer will seem to ignore you.

Attempting to go too fast will result in the slowest speed being selected and vice-versa (look at the instructions at E5E and E61).

The computer will halt the simulation and return to the monitor if the Turing machine enters a HALT state or if the tape head tries to run off the ends of the tape.

TURING MACHINE SIMULATOR

Execute from 0C50

```
START CALL INIT        C50  CD 10 0E
      PRINT            C53  EF 1F 4E 2E
                            20 53 2E 20
                            44 2E 3F 1F 00  'N/LN. S. D.?N/L'
      CALL CHIN        C60  CD 3E 00        Check for new diagram.
      CMP 'Y           C63  FE 59
      JRNZ TAPEI         5  20 3E           If not a new diagram.
      PRINT            C67  EF 1E 4E 6F
                            2E 20 6F 66
                            20 53 74 61
                            74 65 73 3F     'C/SNo. of states?N/L'.
                            1F 00
      CALL LINE        C79  CD F3 0D        to read reply.
      LD B,A             C  47
      PUSH BC            D  C5              hide result.
      XOR A              E  AF              parameter for DISPLAY.
      CALL DISPLAY       F  CD 78 0E        display first prompt.
NEXT  CALL LINE        C82  CD F3 0D        to read reply.
      EXX                5  D9              ENLIST uses the other
                                            registers.
      CALL ENLIST        6  CD A3 0E        to do the work.
      EXX                9  D9
      JRNC NEXT          A  30 F6           carry flag set at end. (FID).
      LD HL,0F50H        C  21 50 0F        top of state diagram.
      PUSH HL            F  E5
      POP IX            90  DD E1           or LD IX,HL.
      16(NOP)            2  00 00. . .      padding.
      PRINT            CA2  EF 1E 00        'C/S'.
TAPEI LD HL,BCAH         5  21 CA 0B
      LD (C18),HL        8  22 18 0C        move cursor.
      PRINT              B  EF 20 4E 6F
                            77 20 77 72
                            69 74 65 20
                            79 6F 75 72
                            20 74 61 70
                            65 3B 20 3C
                            3D 4C 2C 20     'Now write your tape;
                            3E 3D 52 00     =L,  =R'.
      LD HL,80AH       CCB  21 0A 08        top of screen.
      LD DE,0010H        E  11 10 00
      LD B,0FH          D1  06 0F           line count.
OUTER PUSH BC            3  C5
      LD B,30H           4  06 30           column count.
INNER LD (HL),30H        6  36 30           write '0' over all screen.
      INC HL             8  23
      DJNZ INNER         9  10 FB
      ADD HL,DE          B  19              skip over dead screen.
      POP BC             C  C1
      DJNZ OUTER         D  10 F4
      LD HL,9CAH         F  21 CA 09        clear a line for tape head.
      LD B,30H          E2  06 30           column count.
CLINE LD (HL),20H        4  36 20           write ' '.
      INC HL             6  23
      DJNZ CLINE         7  10 FB
      LD HL,9E2H         9  21 E2 09        initial head position.
      LD (HL),'v         C  36 76           'v is the head.
      LD (0BFAH),HL      E  22 FA 0B        record head position.
      PRINT             F1  EF 20 09 00     '—>'.
TAPE  LD DE,40H          5  11 40 00        40 is difference between head
                                            position and symbol read.
      CALL CHIN          8  CD 3E 00        to read instructions.
      CMP '$             B  FE 24
      JRZ GO             D  28 29           start simulation.
      CMP '>             F  FE 3E
      JRZ SWRT         D01  28 10           set direction right.
      CMP '<             3  FE 3C
      JRZ SWLT           5  28 10           set direction left.
      ADD HL,DE        D07  19              HL to read symbol.
      LD (HL),A          8  77              write A there.
      SUB HL,DE          9  ED 52           restore HL.
MVE   CALL MOVE          B  CD CA 0D        to move head.
      LD (BFAH),HL       E  22 FA 0B        update head position record.
```

```
      JR TAPE           11  18 E2           next instruction.
SWRT  LD B,0H            3  06 00           B is direction flag.
      JR MVE             5  18 F4
SWLT  LD B,01H           7  06 01
      JR MVE             9  18 F0
HALT  PRINT              B  EF 48 61 6C
                            74 65 64 2E
                            1F 00           'HALTED.N/L'.
      JP PARSE          25  C3 86 02
GO    LD A,(IX+2)        8  DD 7E 02        IX points to top of current
                                            state; A now has no. of
                                            symbols.
      OR A               B  B7              check for halt state.
      JRZ HALT           C  28 ED
      CALL MONIT         E  CD 40 0E        for control instructions.
      CALL SETDIRN      31  CD E8 0D        dir'n. of motion may have
                                            changed.
      LD HL,(BFAH)       4  2A FA 0B        point HL at head.
      CALL MOVE          7  CD CA 0D        and move it.
      LD (BFAH),HL       A  22 FA 0B        update head position record.
      CALL DELAY         D  CD 00 0E
      LD C,40H          40  0E 40           find symbol under head.
      ADD HL,BC          2  09
      LD A,HL            3  7E
      PUSH IX            4  DD E5           set up registers for CPIR to
      POP HL             6  E1              find if read symbol is
                                            recognised.
      2(INC HL)          7  23 23
      LD C,(HL)          9  4E
      LD D,C             A  51
      INC HL             B  23
      CPIR               C  ED B1
      JRNZ GO            E  20 D8           loop if symbol not recognised
      SUB HL,BC         50  ED 42           now find new state.
      LD A,D             2  7A              D had no. of symbols.
      DEC A              3  3D
      RLC A              4  07
      ADD A,L            5  8D
      LD L,A             6  6F              HL now points to DNST.
      LD D,(HL)          7  56              D has DNST.
      INC HL             8  23
      LD BC,40H          9  01 40 00
      LD A,(HL)          C  7E              A now has OPSY.
      LD HL,(BFAH)       D  2A FA 0B
      ADD HL,BC         60  09              HL now points under head.
      LD (HL),A          1  77              write OPSY.
      LD A,D             2  7A              now move to new state.
      LD H,0FH           3  26 0F
      SLA A              5  CB 27           hide direction information.
LOOP  LD E,(IX+0)        7  DD 5E 00        STID.
      SLA E              A  CB 23
      CP A,E             C  BB              A has DNST.
      JRZ FOUND          D  28 08
      LD L,(IX+1)        F  DD 6E 01        next in chain.
      PUSH HL           72  E5
      POP IX             3  DD E1
      JR LOOP            5  18 F0
FOUND 9(NOP)             6  00 00. . .      for non searching diagrams.
      JR GO            D80  18 A6
      3(NOP)             2  00 00 00        padding.
MOVLINE
      PUSH BC          D85  C5              called by MOVE if head
      PUSH HL            6  E5              moves off edge of screen.
      LD DE,HL           7  54 5D           DE points to line destination.
      LD C,40H           9  0E 40
      DJNZ RMOV1         B  10 1C           if head moving right.
      SUB HL,BC          D  ED 42           point HL to line to be moved.
      LD B,30H           F  06 30           column count.
LP1   LDI               91  ED A0           move one square.
      DEC HL             3  2B
      LD (HL),20H        4  36 20           clear old line.
      INC HL             6  23
      DJNZ LP1           7  10 F8
      LD BC,80AH         9  01 0A 08        now check if off screen at
                                            top.
```

```
        POP HL           C    E1
        SUB HL,BC        D    ED 42
        LD A,H           F    7C
        OR A,L           A0   B5           i.e. if HL = 0.
        JRZ BANG         1    28 23        if it's run off.
        ADD HL,BC        3    09
        LD BC,FFF0H      4    01 F0 FF     = –10H the change in head
                                           position.
        JR OUT           7    18 1A
RMOV1   INC B            9    04           B now = 0.
        ADD HL,BC        A    09           much as when head moves
                                           left.
        LD B,30H         B    06 30
RLP     LDD              D    ED A8        move square.
        INC HL           F    23
        LD (HL),20H      B0   36 20        clear old line.
        DEC HL           2    2B
        DJNZ RLP         3    10 F8
        LD BC,B79H       5    01 79 0B     check if off at bottom.
        POP HL           8    E1
        SUB HL,BC        9    ED 42
        LD A,H           B    7C
        OR A,L           C    B5
        JRZ BANG         D    28 07
        ADD HL,BC        F    09
        LD BC,10H        C0   01 10 00     the change in head positon.
OUT     ADD HL,BC        3    09
        POP BC           4    C1
        RET              5    C9
BANG    POP AF           6    F1           tidy up stack.
        JP HALT          7    C3 1B 0D     and halt.
MOVE    LD HL,20H        A    36 20        clear old head.
        LD A,L           C    7D
        AND 3FH          D    E6 3F        check for end of line, first
                                           stage.
        SUB 0AH          F    D6 0A
        DJNZ RMOV        D1   10 09        B has direction flag.
        PUSH AF          3    F5           hide flags.
        INC B            4    04           restore B.
        POP AF           5    F1           restore flags.
        CALLZ MOVLINE    6    CC 85 0D     if off at left end.
        DEC HL           9    2B
        JR XIT           A    18 09
RMOV    SUB 2FH          C    D6 2F        second stage check for eol.
        PUSH AF          E    F5
        INC B            F    04
        POP AF           E0   F1
        CALLZ MOVLINE    1    CC 85 0D     if off at right end.
        INC HL           4    23
XIT     LD (HL),76H      5    36 76        replace tape head ('V).
        RET              7    C9
SETDIRN
        LD B,1H          8    06 01
        LD A,(IX+0)      A    DD 7E 00     STID.
        RLA              D    17           direction bit into carry.
        JRC SKIP         E    38 01
        DEC B            F0   05           B now = 0.
SKIP    RRA              1    1F
        RET              2    C9
LINE    CALL INLINE      DF3  CD DB 01     read a line.
        LD DE,B4BH       6    11 4B 0B     = where it is now.
        CALL NEXNUM      9    CD 5A 02     translate it.
        LD A,(C13)       C    3A 13 0C     and put the number in A.
        RET              F    C9
DELAY   PUSH HL          E00  E5
        LD B,1H          1    06 01        fast delay; MONIT alters this
                                           value.
        LD DE,FFFFH      3    11 FF FF     = –1H.
CYCLES
        LD HL,2393H      6    21 93 23
DOWN    ADD HL,DE        9    19
        JRC DOWN         A    38 FD
        DJNZ CYCLES      C    10 F8
        POP HL           E    E1
        RET              F    C9
INIT    EXX              10   D9           set up other registers for
                                           enlist.
        LD C,00H         1    0E 00
        LD HL,F50H       3    21 50 0F
        PUSH HL          6    E5
        POP IX           7    DD E1
        EXX              9    D9
        EX               A    08           set auto/manual flag.
        LD A,FFH         B    3E FF
        EX               D    08
        RET              E    C9
        21(NOP)          F    00 00. . .   for non searching facility.
MONIT   EX               E40  08           check auto/man. flag.
        OR A             1    B7
        JRNZ MANUAL      2    20 20        if A = 0 then auto.
        EX               4    08           hide flag.
        CALL KBD         5    CD 69 00     KBD does not loop.
        RET NC           8    D0           no key found.
```

```
        CMP 'M           9    FE 4D
        JRZ SETMAN       B    28 0C        set to manual please.
        LD HL,E02H       D    21 02 0E     address of the byte to change
                                           speed.
        CMP 'F           50   FE 46
        JRZ UPSPD        2    28 0A
        CMP 'S           4    FE 53
        JRZ DNSPD        6    28 09
        RET              8    C9           character not recognised.
SETMAN
        EX               9    08           get flag.
        LD A,FFH         A    3E FF
        JR MANUAL        C    18 06
UPSPD   RRC (HL)         E    CB 0E        rotate to shorten delay loop.
        RET              30   C9
DNSPD   RLC (HL)         1    CB 06        rotate to lengthen delay loop.
        RET              3    C9
MANUAL
        EX               4    08           hide flag.
        PRINT            5    EF 3A 00     ':'.
        CALL CHIN        8    CD 3E 00     CHIN loops till key found.
        LD H,A           B    67
        PRINT            C    EF 1D 00     'B/S'.
        LD A,H           F    7C
        CMP 'A           70   FE 41
        RET NZ           2    C0           only 'A matters.
        EX               3    08           new flag.
        LD A,00H         4    3E 00
        EX               6    08
        RET              7    C9
DISPLAY
        PUSH HL          E78  E5           hide main registers.
        PUSH DE          9    D5
        PUSH BC          A    C5
        LD HL,E8FH       B    21 8F 0E     top of prompts table.
        RLC A            E    07           A has ID of prompt required.
        RLC A            F    07
        ADDL             80   85
        LD L,A           1    6F           HL points to required prompt.
        LD BC,04H        2    01 04 00     character count.
        LD DE,BCAH       5    11 CA 0B     destination of prompt.
        LDIR             8    ED B0        transfer the characters.
        POP BC           A    C1           recover registers.
        POP DE           B    D1
        POP HL           C    E1
        AND A            E8D  A7           clear carry flag.
        RET              E    C9
PROMPTS TABLE            F    53 54 49 44  STID.
                         93   4E 52 53 59  NRSY.
                         7    49 50 53 59  IPSY.
                         B    44 4E 53 54  DNST. .
                         F    4F 50 53 59  OPSY.
ENLIST  CMP FFH          EA3  FE FF        check for error input.
        JRZ ERR          5    28 78
        TEST Cb1         7    CB 49        C contains flags on the state
                                           of enlist, bit 1 set for IPSY,
                                           DNST & OPSY states.
        JRNZ TRIPLES     9    20 23
        TEST Cb0         B    CB 41        bit 0 set if expecting NRSY.
        JRNZ STATES      D    20 08
        LD (HL),A        F    77           expecting STID, put HL.
        INC HL           B0   23
        INC C            1    0C           i.e. set bit 0.
        LD A,C           2    79           i.e. with 1 to get prompt no.1
        CALL DISPLAY     3    CD 78 0E     when DISPLAY is called.
        RET              6    C9
STATESLD D,A             7    57           first work out where next
                                           state will start; hide NRSY in
                                           D.
        SLA A            8    CB 27        calculate in A the 1.s. byte of
        ADD A,D          A    82           the address of the next state's
        ADD 2H           B    C6 02        start; the m.s. byte will be 0F.
        ADD A,L          D    85
        LD (HL),A        E    77           use this as pointer.
        LD E,L           F    5D           save 1.s. byte of address of
                                           pointer it may have to be
                                           changed to make chain into
                                           ring.
        INC HL           C0   23
        LD (HL),D        1    72           NRSY stored next.
        INC HL           2    23
        INC C            3    0C           i.e. set bit 1.
        LD B,D           4    42
        LD A,D           5    7A
        OR A             6    B7           if it's a halt state there are no
        JRZ ENDL         7    28 3A        triples.
        LD A,C           9    79           i.e. with 2.
        CALL DISPLAY     A    CD 78 0E     for prompt no. 2 = IPSY.
        RET              D    C9
TRIPLES
        TEST Cb5         E    CB 69        bit 5 clear if expecting IPSY.
        JRNZ NOT1ST      D0   20 0F
        SET Cb5          2    CB E9
```

```
        LD (HL),A        4    77            write IPSY in table, then
        LD A,D           5    7A            advance HL to point to top
        SLA A            6    CB 27         of DNST/OPSY pair. IPSY's
        ADD A,L          8    85            in table for lookup.
        SUB A,B        ED9    90
        LD L,A           A    6F            HL now points to address of
                                            DNST.
        LD A,3H          B    3E 03         parameter for DISPLAY.
        CALL DISPLAY     D    CD 78 0E
        RET             E0    C9
NOT1ST
        TEST Cb4         1    CB 61         bit 4 set if expecting OPSY.
        JRNZ LAST        3    20 0A
        SET Cb4          5    CB E1
        LD (HL),A        7    77
        INC HL           8    23            now points to home for
                                            OPSY.
        LD A,4H          9    3E 04
        CALL DISPLAY     B    CD 78 0E
        RET              E    C9            that one was easy.
LAST    LD (HL),A        F    77            store OPSY then restore HL
        LD A,B          F0    78            to point to next place in
                                            IPSY table.
        ADD L            1    85            B has no. of symbols left to
                                            read.
        LD L,D           2    6A            D has total no. of symbols.
        SLA L            3    CB 25
        SUB A,L          5    95
        LD L,A           6    6F
        DJNZ MORE        7    10 02         check if more symbols for
                                            this state.
        JR ENDL          9    18 08         if list has ended.
MORE    LD C,3H          B    0E 03         set two flags.
        LD A,2H          D    3E 02
        CALL DISPLAY     F    CD 78 0E
        RET            F02    C9
ENDL    POP AF           3    F1            don't lose return address.
        POP BC           4    C1            B now has no. of STATES.
        DJNZ NMORE       5    10 02
        JR ENDST         7    18 10         if no more states.
NMORE   PUSH BC          9    C5            hide BC.
        PUSH AF          A    F5            and put back return address.
        LD A,(IX+1)      B    DD 7E 01      l.s. byte of top of next state.
        LD L,A           E    6F            point HL there.
        PUSH HL          F    E5            and LD IX,HL.
        POP IX          10    DD E1
NEW     LD C,0H          2    0E 00         clear flags.
        LD A,C           4    79
        CALL DISPLAY     5    CD 78 0E
        RET              8    C9
ENDST   PUSH AF          9    F5            save return address.
        LD L,E           A    6B            complete data structure by
        LD (HL),50H      B    36 50         pointing from last state to
        SCF              D    37            top of first state at F50.
                                            Carry flag set for main pgm.
        RET              E    C9
ERR     PUSH IX          F    DD E5         point HL to top of state (in
                                            IX).
        POP HL          21    E1
        LD A,L           2    79            check for return to previous
                                            state.
        OR  A            3    B7
        JRNZ NEW         4    20 EC         clear state and start again.
        PUSH BC          6    C5            hide BC.
        LD BC,F50H       7    01 50 0F      don't step back beyond 0F50.
        SUB HL,BC        A    ED 42
        POP  BC          C    C1            recover BC.
        LD A,L           D    7D
        OR  H            E    B4
        JRZ CRASH        F    28 11         if HL-0F50=0
        POP AF          31    F1            return address.
        POP BC           2    C1            state count must be
                                            incremented.
        INC B            3    04
        PUSH BC          4    C5
        PUSH AF          5    F5
        INC D            6    14            now point HL to top of
        LD C,D           7    4A            previous state.
        SLA  C           8    CB 21
        LD A,L           A    7D
        SUB  C           B    91
        SUB  D           C    92
        LD L,A         F3D    6F
        PUSH HL          E    E5            and put HL in IX.
        POP IX           F    DD E1
        RET             41    C9
CRASH   PRINT            2    EF 1F 4E 4F   if step back too far:
                              21 1F 00      'N/LNO!N/L'.
        JP PARSE         9    C3 86 02      re-enter monitor.
        4(NOP)           C    00 00 00 00   padding.

START OF STATE DIAGRAM F50
```

MAIN PROGRAM	CALLS	CALLS
START 0C50	INIT 0E10	
	LINE 0DF3	
	DISPLAY 0E78	
	ENLIST 0EA3	DISPLAY 0E78
TAPEI 0CA5		
CLINE 0CE4		
TAPE 0CF5	MOVE 0DCA	MOVLINE 0D85
GO 0D28	MONIT 0E40	
	SETDIRN 0DE8	
	MOVE 0DCA	MOVLINE 0D85
	DELAY 0E00	

Summary Of Subroutine Functions

INIT	Sets up registers and flags.
LINE	Reads a hexadecimal number from the keyboard, displays it, and puts its value in A.
DISPLAY	Takes a number in A and displays on the screen a 4 letter message which depends on A's contents. Numbers greater than 4 give strange results.
ENLIST	Uses the alternative register set, and creates a Data Structure representing the Turing Machine's State Diagram.
MOVE & MOVLINE	Take the position of the tape head in HL, the direction of required movement in B and move the head one square, perhaps skipping a line.
SETDIRN	Sets B according to whether the state pointed to by IX is R or L.
MONIT	Reads control characters from the keyboard and can change the speed of the delay loop. In automatic mode it will not wait if no key is pressed.
DELAY	Executes a delaying loop. One loop parameter can be altered by MONIT.

QUARTZ LCD 5 Function

Hours, mins., secs, month, date, auto calendar, back light, quality metal bracelet

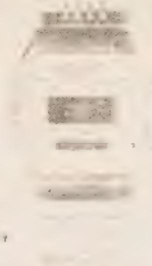
£6.65

Guaranteed same day despatch.

Very slim, only 6mm thick

M1

SOLAR QUARTZ LCD 5 Function



Genuine solar panel with battery back-up
Hours, mins., secs.
Day, date
Fully adjustable bracelet
Back-light
Only 7mm thick

£8.65

Guaranteed same day dispatch

M2

QUARTZ LCD 11 Function SLIM CHRONO

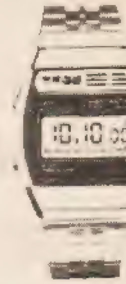
6 digit 11 functions
Hours, mins., secs, day, date, day of week
1/100th, 1/10th secs
10X secs, mins.
Split and lap modes
Back light, auto calendar. Only 8mm thick
Stainless steel bracelet and back
Adjustable bracelet
Metac Price

£10.65 Thousands sold!

Guaranteed same day dispatch

M3

QUARTZ LCD ALARM 7 Function



Alarm
Hours, mins., secs
Month, date, day
6 digits 3 flags plus continuous display of day and date or seconds
Back-light
Only 9mm thick

£12.65

Guaranteed same day dispatch

M4

MULTI ALARM 6 Digits 10 Functions

- ★ Hours, mins., secs
- ★ Month, date, day
- ★ Basic alarm
- ★ Memory date alarm
- ★ Timer alarm with dual time and 5 country zone
- ★ Back light
- ★ 8mm thick

£18.65

M5

FRONT-BUTTON ALARM Chrono Dual Time

6 digits, 5 flags.
22 functions
Constant display of hours and mins. plus optional seconds or date display
AM/PM indication
Month, date
Continuous display of day
Stop-watch to 12 hours 59.9 secs in 1/10 second steps
Split and lap timing modes
Dual time zones
Only 8mm thick
Back-light

Fully adjustable open bracelet.

£22.65

Guaranteed same day dispatch

M6

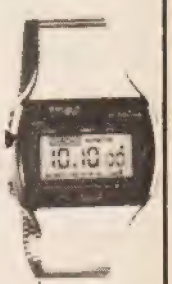
SOLAR QUARTZ LCD Chronograph with Alarm Dual Time Zone Facility

6 digits, 5 flags
22 functions
Solar panel with battery back-up
6 basic functions
stop-watch to 12 hours 59.9 secs in 1/10 sec. steps
Split and lap timing modes
Dual time zones
Alarm
9mm thick
Back-light
Fully adjustable bracelet

£27.95

M7

ALARM CHRONO with 9 World Time Zones

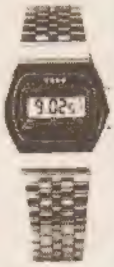


- ★ 6 digits, 5 flags
- ★ 6 basic functions
- ★ 8 further time zones
- ★ Count-down alarm
- ★ Stop-watch to 12 hours 59.9 secs in 1/10 sec. steps
- ★ Split and lap timing modes
- ★ Alarm
- ★ 9mm thick
- ★ Back-light
- ★ Fully adjustable bracelet

£29.65

M8

SOLAR QUARTZ LCD Chronograph



Powered from solar panel with battery back up
6 digit, 11 functions
Hours, mins., secs., day, date, day of week
1/100th, 1/10th secs
10X secs., mins.
Split and lap modes
Back light. Auto calendar. Only 8mm thick
Stainless steel bracelet and back
Adjustable bracelet
Metac Price

£13.65 Guaranteed same day dispatch

M9

QUARTZ LCD

Lady's Day Watch only 25×20mm and 6mm thick
Hours, minutes, seconds, day, date backlight and auto calendar.
Elegant metal bracelet in silver or gold fully adjustable to suit very slim wrists.
State colour preference.

£9.95 Guaranteed same day despatch

M15

QUARTZ LCD

Lady's Fashion Watch. Elegant bracelet in bronze/gold finish or silver colour. Hours, minutes, seconds, day, date, backlight and auto calendar. Adjustable for the slimmest of wrists.

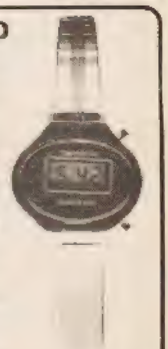
State colour preference.

£14.95

Guaranteed same day despatch

M17

QUARTZ LCD



Lady's Cocktail Watch
Highly functional watch which also suits those special occasions. Beautifully designed with a very thin bracelet which retains strength as well as elegance. Hours, mins., secs., day, date backlight and autocalendar.
Bracelet fully adjustable to suit slim wrists.
State gold or silver finish.

£19.95

Guaranteed same day despatch

M18

HANIMEX Electronic LED Alarm Clock

Features and Specification:

Hour minute display. Large LED display with p.m. and alarm on indicator. 24 Hours alarm with on off control. Display flashing for power loss indication. Repeatable 9-minute snooze. Display bright dim modes control. Size: 5.15" x 3.93" x 2.36" (131mm x 11mm x 60mm).
Weight: 1.43 lbs (0.65 kg).

£10.20 Thousands sold!

Mains operated

Guaranteed same day despatch

M13

EXECUTIVE ALARM WATCH

6 functions plus alarm. Conference signal. 5 minute snooze alarm. Conference signal sounds 4 secs. before main alarm to give advance warning and option to cancel. Snooze sounds 5 mins after main alarm and is always preceded by the conference signal

£14.95

M60

MACY QUARTZ ANALOGUE

Automatic calendar day and date, infinite bracelet. This man's watch has elegance as well as the robust appearance provided by a watch with traditional features. Accuracy is provided by a quartz crystal powered by a long life miniature battery

£24.95

M21

Metac price breakthrough for an Alarm Chronograph with Dual Time Only £18.95

OUTSTANDING FEATURES

- ★ **DUAL TIME.** Local time always visible and you can set and recall any other time zone (such as GMT). Also has a light for night viewing
- ★ **CALENDAR FUNCTIONS** include the date and day in each time zone
- ★ **CHRONOGRAPH/STOPWATCH** displays up to 12 hours, 59 minutes and 59.9 seconds. On command, stopwatch display freezes to show intermediate (split/lap) time while stopwatch continues to run. Can also switch to and from timekeeping and stopwatch modes without affecting either's operation
- ★ **ALARM** can be set to any time within a 24-hour period. At the designated time, a pleasant, but effective buzzer sounds to remind or awaken you!

Guaranteed same day despatch **M16**

HOW TO ORDER

Payment can be made by sending cheque, postal order, Barclay, Access or American Express card numbers. Write your name, address and order details clearly, enclose 40 pence per single item for post and packing or the amount stated in the advert. All products carry 1 year written guarantee and full money-back 10 day reassurance. Battery fitting and electronic calibration service is available to customers at any Metac shop. All prices include VAT currently at 15%.

Metac Wholesale:
Trade enquiries — send for a complete list of prices for all the goods advertised plus many more not shown, also minimum order details.
Telephone orders. Credit card customers can telephone orders direct to Daventry (03272) 76545 or Edgware Road 01-723 4753 24 hours a day.

Service Enquiries: 03272-77659
CALLERS WELCOME. Shops open 9.30am-6.00pm.

Metac ELECTRONICS & TIME CENTRES

North & Midlands
67 High Street, DAVENTRY
Northamptonshire
Telephone: 03272 76545

South of England
327 Edgware Road
LONDON W.2
Telephone: (01) 723 4753

Ian Sinclair

A first time buyers review of one of the top selling home computers

This is a report on how a "beginner" bought a TRS-80. Beginner? I'll admit to having programmed before, but that was using FORTRAN, and it was before BASIC was invented. I'll admit cheerfully to having a certain amount of experience of MPU's and other hardware, as the columns of this magazine witnesseth. But as far as BASIC programming and the use of the microcomputer goes, I'm a beginner. Or I was. . . .

Why did I wait so long? Being an old hand in the electronics business, and having seen what happened to the prices of calculators, I felt that prices had to drop. In addition, I had been in the States in 1977 and seen the prices which were being charged there; I've kept in touch with their prices ever since! Next item was that I didn't want to buy a computer to play games, if the thing couldn't earn its keep, it wouldn't be bought. Third item was that I already had a spare TV and a cassette recorder, and it seemed a bit pointless to buy a computer which included these items all over again.

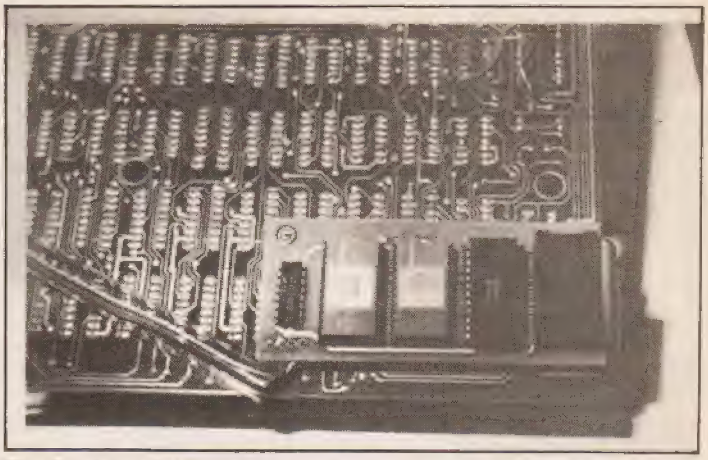
A Problem Waiting For Solution

Time came and went, and I kept reading specifications. By this time, early this year, several applications were starting to crystallise. I spend a bit of time writing textbooks, and one of the most time-consuming parts of the writing process is indexing the page proofs. At present, it involves underlining, on the page proofs, each word that is needed in the index, and writing that word, with its page number, on a sheet which is reserved for its initial letter. The words then have to be arranged in strict alphabetical order, and where the same index word occurs on several pages, the page numbers grouped alongside the word. It's a messy bit of donkey-work, with 26 sheets of paper flying about, and the sorting of words into strict alphabetical order taks a surprising amount of time. This task looked like a prime target for the use of a computer.

The other main item was the keeping of drawings files. I have a collection of drawings, copies which by now must amount to about five hundred items. They are filed under subject headings, but to find the one I want, I may have to look through more than one index. This looked like another target, provided the computer could handle files on cassette, possibly going to disc later. Thirdly came the inevitable financial application, keeping track of who paid what, and when, for what, and when, and if not, why not.

The tentative specification, then, was for plenty RAM, around 16K, and no need for frills like colour graphics (or any graphics, for that matter) provided there was plenty of string-handling capability. At the time I had reached this stage, prices were still well above the level at which I judged the thing could pay its own way, but the TRS-80 Level-2 looked a useful contender.

The traditional QWERTY layout is retained on the TRS-80 and it also offers a separate numeric pad.



The Extended BASIC ROMs on their sub-chassis for level two.

Price Cut Equals Problem Solution

That situation suddenly changed when Comp of Barnet decided to bring a touch of realism to U.K. prices by offering the 16K TRS-80 Level-2 at £399 (plus Very Atrocious Tax). The Level-2 has very much greater string-handling capability than the Level-1, and 16K of memory is a respectable amount. In addition, being dynamic memory means that the power supply doesn't have to be a miniature version of Battersea Power Station, glowing a dull red; and adding yet another 16K later doesn't strain the power requirements too much. Were there any snags? Cassette compatibility with other machines didn't worry me too much — most of my programs are of interest only to me, and when you're compatible with the world's biggest-selling microcomputer, that's compatible enough. Similarly, I don't worry overmuch about not having the S-100 bus (should anyone?). Standard connections and systems are essential if you don't know anything about the hardware; they become less important if you know what's going on, and hardly matter at all if every shop stocks interfaces and hardware for your system.

All in all, it looked like a good case for getting on a train at Moorgate and, when BR could find a train to go all the way, off at New Barnet. A few yards from the platform, in Station Road, TRS-80's were changing hands rapidly. Not, perhaps, like Tottenham Court Road on a wet Saturday afternoon, but fast enough to bring a smile to their faces. I saw mine flash up its memory size, slapped it back in the box, and went off with it.

How do they do it? Officially, I don't know, but there's a clue on the back of my TRS-80. It has the Radio Shack LEVEL 2 4K label on, which suggests that the merry men of Barnet probably ship them in as 4K units, add another 12K of RAM and, bingo, 16K. At the same time, they sort out the old US/UK TV standards bogey, so that the picture on a TV receiver is absolutely rock steady. Since the price includes the power supply and the video modulator, your unit is complete and ready to go as soon as you take it out of the box and connect a plug. They have, need I say, installed a 240 V transformer. Full marks, chaps, for initiative, friendliness and good soldering.

The Home Stretch

Now for that can't-wait-to-get-it-home feeling. With all the hardware unpacked, I found some of the very useful software that comes with the TRS-80. There was, to start with, the LEVEL-1 BASIC MANUAL. Now I've read quite a few books on BASIC, most of them, in fact, but the TRS-80 LEVEL-1 manual knocks them all into a cocked hat. It's a model of clarity, logically set out, well written, and a sheer joy to read. There are no comic strip sequences, no learned obscurity, just sheer good explanation with lots of examples. The name to remember is Dr. David Lien — I'll be very happy if my writing ever becomes half as good as his.

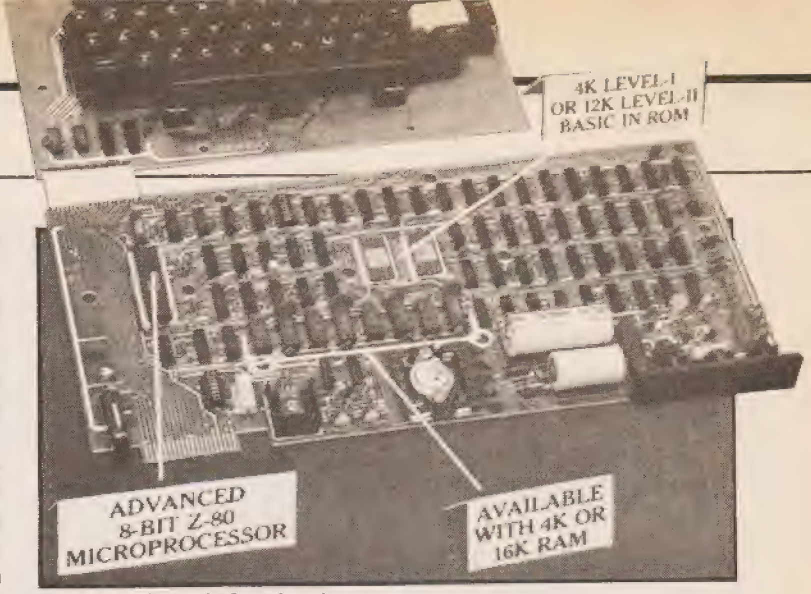
There's just one small snag — quite a lot of the LEVEL 1 material doesn't apply to LEVEL-2, as you discover on the first page when you try to print the memory size. On LEVEL-1, the abbreviation PM is used to do this; LEVEL-2 requires the words to be typed out a bit more fully, as PRINT MEM. Differences of this sort don't really cause too much bother, once you've met the first few.

The LEVEL-2 manual is also included with the outfit. This isn't intended for beginners, it's more of a dictionary to the LEVEL-2 dialect of BASIC than a textbook. As far as I can judge, LEVEL-2 BASIC is pretty standard, with several of the little luxury touches, such as TAB, which delight the keen typist. The software package is completed by three cassettes, one with games (Backgammon and Blackjack), one with systems programs in machine-code (Z-80, if you didn't know) and one blank. All in all, you get a lot for your money.

The real test, though, is of its ability to do the job it was bought for. First indications were that it would. I could get only indications at first, because I need rather specialised programs, which were going to need time to write. I started to run small adaptations of standard programs, and also to run across the points which are not covered completely by any manuals — more of these later.

Loaded To Save It

The first thing I knew I had to sort out was cassette loading and saving, so that a precious program developed with sweat and tears (no blood, because there are no sharp edges) over days of effort could be saved and used. Just as important, a program which was not fully debugged could be saved for a later effort without all the hassle of keying in again. I type fairly quickly, but it's no fun to type in a really long program. Intensive reading about this caused me some worries, because there were endless reports of problems with TRS-80 cassette loading and dumping. My own problem number one was that themicrophone and control inputs of my cassette recorder (Normende 391) were at a 7-pin DIN input, whereas the TRS80 uses sensible 3.5 mm and 2.5 mm jack plugs. Out came the soldering-iron and the metal-basing tools, and after a bit of work an adaptor had been created. Two of the plugs



from the TRS-80 now go into sockets on a small box, whose output cable sports a 7-pin DIN plug — the remaining jack-plug goes into a headphone jack socket on the recorder. Just as a touch of luxury, I added a switch so that I could recover control of the recorder motor when the computer had switched it off-it makes rewinding much easier. The first priority was to get replay going, so the Blackjack cassette came in for its only likely use.

If you're not familiar with the system, the cassette loading scheme for the TRS-80 is quite neat. An asterisk appears at the top right of the monitor screen to show that a program is present on a cassette, and a second asterisk flashes to show that a program is loading. If you've selected a labelled program ("A", "B", "5", etc.) then any programs you pass on the way to your selected one are also indicated on the right hand side. It works beautifully, once you know what you are doing — but that can take time!

Pillock's progress went something like this. I set the playback volume low, popped in the cassette, pressed "play", entered CLOAD on the computer, and watched. Nothing happened, no asterisks, volume setting too low. Rewound the tape, set the volume up a bit, went through the ritual again. This time, asterisks appear, but no blinking and no program loaded. Rewind, reset volume, try again — several times. At this stage I started to believe that cassette loading can damage your health; so I had to turn away for a time and re-engage the brain. Surely these ** asterisks were still on the screen while I was rewinding the cassette? They must be latched, so I have to cancel them on each run. A quick fumble through the manual reveals the presence of a reset button under the expansion flap on the rear left side. This causes recovery from the cloading condition (sounds painful), and the asterisks vanish when the next CLOAD is keyed in and entered. Ho, ho and try again. After two more runs, the cassette was loaded. There was slight corruption on the data (would you believe 100 FPB * VT @GH' SKLGH RMP?) so I upped the volume control a little bit, reset and tried again. It loaded perfectly, and loading has given me no trouble since. That lot was enough for one day!

Converging On A Program

Next spare day was try-a-program day. I have already a collection of program listings, but I wanted to write one of my own to check two things — my understanding of BASIC so far, and the convergence of the colour receiver I was using as a monitor. The first seemed OK — the program is shown in Fig. 1; it simply writes the character of your choice in lines all over the screen. I had suspected that the set had lousy convergence, and I wasn't wrong. Furthermore, being a pre-PIL receiver, the convergence was unstable. Ten minutes of twiddling could get reasonable white letters with just a few touches of colour at the edges, but by the time the back was on again, and certainly at the next switch-on, it was as bad as ever, with ghostly red letters at the side of the off-white ones. Mental note, get a black/white receiver, or go the whole hog and nip back to Barnet for a monitor. I don't need colour — the receiver just happened to be there. A little inner voice then says "Chicken! Call yourself an engineer?" OK, I'll try again.

The important thing now was to save the program on a cassette to check the recording process. I typed in CSAVE "A", jabbed enter (it's not RETURN on the TRS-80) and watched the recorder start. Yes, it would have helped if I had pressed the "record" key as well. Back to square one, and repeat. This time it ran — then the 'phone rang and that was the end of work for the day.

Day three started with loading the convergence program back in — or trying to. It wouldn't load at any setting, and no amount of juggling with the controls would give anything but gibberish loaded. By this time, I once again had to get on with something else, and there was no more spare time that day. Who says computers save time?

On day four, I decided to be calm and rational. I tried the Blackjack program again at the original loading settings I had found. It loaded perfectly, but the convergence program didn't. Since the computer couldn't tell me what was going on, I took the earphone plug out and listened to each program in turn. No doubt about it, the convergence was a much fainter and noisier program. We obviously had recording problems. I tried several more CSAVE's of a short program, one, and only one, of which was perfect. The first instinct was to open the box. At that point I remembered the Second Law of the Marquis de Sod (Sod's Law, you know) which says that when you have problems with a complex bit when it's a simple bit that's gone wrong. When your computer doesn't record on cassette, you naturally blame the computer. The fault was, in fact, in my adaptor. The tip of the TRS-80 jack plug is fractionally longer than the jack plugs I normally use, and it was touching — just — the earth wire in the socket. That obviously didn't do the signal any good at all; and I was glad I didn't take the TRS-80 apart. Quite apart from anything else, if I can short out a jack plug, what could I do to 16K or RAM chips? After that small point was sorted out, every save and load worked perfectly. Two small points might be of interest. One is that the recorder has internal motor and microphone switches operated by the DIN plug. The plug I used didn't reach the switch, so a short chunk of matchstick was inserted in the recess of the socket, and inserting the plug then had the desired effect. The other point is that there is an earth loop, because both the microphone and the earphone leads are earthed. I cut away the braid from the earphone lead, which resulted in even better recordings, and I need less volume for reliable playback. The same effect can be achieved with no destruction by making up a socket and plug, connecting only the centre terminals so that the earth is not linked in.

Data Versus Files

With that lot finally sorted out, I could then start on the real work. Details probably aren't of interest to you, unless you are an author with a TRS-80 and in need of an indexing program, but a few conclusions might be, especially if they provoke some solutions to problems. One conclusion is that the TRS-80 is generally pretty well suited to my needs. Another is that cassette file handling is a joke. With only about 248 bytes on each INPUT and PRINT cassette instruction, any reasonable amount of file data takes a long time, and has the reed switch (which operates the recorder motor)

clicking like a demented pool table. The technical manual hints that this reed switch can have a short life in such circumstances, so I make use of that 16K of memory and simply keep my files as DATA statements within the program which enters and reads the files. That way, one cassette load inputs the whole of the program and the data. I agree that the advantage of filing is that one set of data can be used for several purposes, but unless there's a better way, I'll wait for a disc unit when the long-promised Japanese disc drives become available. I believe it must be possible to control that reed switch with a program, but it must either need machine language or a POKE statement. The only reference to machine language in the manual is to loading from cassette, and where and what do you POKE? My problem is that time is limited; much as I would like to experiment, I have to spend time earning a living.

Another conclusion is that there is a considerable variation in the quality of the software which is available. Now I appreciate that software can never be cheap, because of the hours of effort that it takes to prepare and debug, but need it be all that expensive for a standard product which sells in the States for quarter of the price? I have a book called "Some Common Basic Programs", published by that guru of the microcomputing movement, Adam Osborne. It contains some 76 program listings for a price of £6.30 (from L.P. Enterprises). The point is that many of the cassette programs I have seen on offer look uncannily like programs in the book, but at about the same price for each program!

Obviously, no dealer is going to let you have a cassette, or even a listing, on approval, since copying is so easy, and the only way to be sure that a cassette program is good value is to see it running. I wouldn't buy unseen unless I had a great deal of faith in the supplier, built up by experience, or had seen a detailed review of the cassette.

Meantime, between bouts of writing this and other things, my book indexing program is bug-free and is now committed to cassette. I can now enter each word with its page number, and at the last page start an alphabetical sort. When this is complete (and it takes a surprising amount of time), I can type in a letter, and receive an index for this letter, in alphabetical and numerical order. What makes this a bit more than a straightforward sorting operation is that entries are correctly combined. For example, if I have entries of Transistor 47, Transistor 68, Transistor 92; these appear as Transistor 47 68 92, with no repetition of the index word. This is the form I need, and it was worth all the effort.

```
           SCREEN FILL FOR CONVERGENCE

2   CLS
3   INPUT ''TYPE SYMBOL TO BE USED'';A$
4   FOR N = 1 TO 1023
5   PRINT@N , A$;
6   NEXT
```

Fig.1. To run, type RUN, press ENTER. At the first question, type in the character you want to use (the full stop and + signs are the most useful), and press ENTER again.

Conclusion

Half-fun, half-spite, the machine is often called the TRASH-80. For me, it's TRUSTY-80, and I'm very satisfied. Thank you, Tandy Corporation, thank you, Comp. Now about that cheap printer and disc unit . . . .

Tandy Replies:—

Dear Sir,

I would like to thank the Editor for this opportunity to reply to the interesting article on the TRS-80 by Mr. I. Sinclair.

I should like to begin by congratulating Mr. Sinclair on his excellent choice of micro-computer and thank him for the christening "Trusty-80". He has coined in this tiny phrase the feelings of thousands of other satisfied customers.

There are a few points I would like to make regarding his purchase, however, and hope to clarify one or two questions which might have arisen in the minds of our future customers.

It seems that Mr. Sinclair took some time to decide exactly which microprocessor would be best suited for his requirements and, like any sensible person would, looked around to find the best possible price. In doing so however, he appears to have overlooked Tandy Corporation, who – being the manufacturer – would be the obvious choice of supplier for most. Further, Mr. Sinclair did not purchase a 16K Level II TRS-80 as supplied by Tandy Corporation. What he did purchase was a single component part – namely, the 'Keyboard' (also known as the central processing unit) – from a computer component supplier, not an authorised Tandy agent. In terms of capital outlay he paid £399.00 (exclusive of V.A.T.) for one part of a 16K Level II System. TO COMPLETE HIS PURCHASE he would need to buy a 12" black and white video monitor and a cassette recorder with a digital tape counter.

As Mr. Sinclair was already in possession of a cassette recorder – for which he had to build an adaptor box (which led to wasted time and 'he calls himself an engineer') he only needs to purchase a video monitor.

Were it not that Mr. Sinclair had found the resultant picture on his Television receiver was inadequate (a U.H.F. modulator was included in his purchase price) this would not have been necessary. I feel bound to say that we have been advising people against the use of these modulators from the beginning.

Considering all this, we find that it is easier AND CHEAPER – AS ONE MIGHT EXPECT – to purchase the TRS-80 directly from the manufacturer, namely Tandy Corporation.

Apart from these obvious benefits, Tandy Corporation can offer an after sales service second-to-none on a national basis; but not on units imported AND CONVERTED TO U.K. STANDARDS BY UNAUTHORISED AGENTS.

I would like to recommend, therefore, that any future customers learn from Mr. Sinclair's mistakes and purchase direct from an authorised Tandy dealer. I hope that when Mr. Sinclair comes to purchasing his expansion units he will save himself the aggravation and expense he has experienced to date by buying from a Tandy retail outlet.

I should like to conclude by thanking Mr. Sinclair for his fair and kind appraisal of the TRS-80 and hope that he continues to benefit from its use.

Mr. R.A. Heller,
Manager,
Computer Division.

Ian Sinclair

MPU's BY EXPERIMENT

Our series continues with a close look at some more machine code instructions

In part 4 we came across the binary-add instruction ADI (the immediate version of binary-add, that is) and the way that the CY/L and OV status bits are used. We're not completely finished with immediate-loading instructions yet, however, and there are a few rather important ones left. The first one we'll look at this month is the DECIMAL–ADD IMMEDIATE, DAI, coded as 11101100. For decimal addition, the byte in the accumulator and the data byte added to the accumulator are both treated as two-digit BCD numbers. For example, the byte 01010011 is taken as the two BCD digits 0101 (decimal 5) and 0011 (decimal 3), making 53. Some care is needed to ensure that the numbers in the accumulator and in the memory are genuine BCD numbers; 11001010 isn't because its first four bits decode to 12 and its next four bits to 10.

BCD Equals Good News

The ability to carry out BCD arithmetic is rather useful, because it's not the easiest thing in the world to write programs for – Fig. 1 indicates what is needed to add BCD numbers. We can try out two examples, one of which adds simply, like binary numbers, the other of which reveals some of the actions which are taking place inside the microprocessor. After the usual RESET and CANCEL, load up BCD 25 by using the steps 11000100 (LDI) PUSH 00100101 (BCD 25) PUSH. The next steps are 11101100 (DAI) PUSH 00110010 (BCD 32) PUSH. We can view the answer by the usual 11001000 PUSH 00000001 PUSH routine. This gives the answer 01010111 which is BCD 57, the correct answer. In this example, the BCD addition has been indistinguishable from binary addition, because there has been no carry from one group of four to the next.

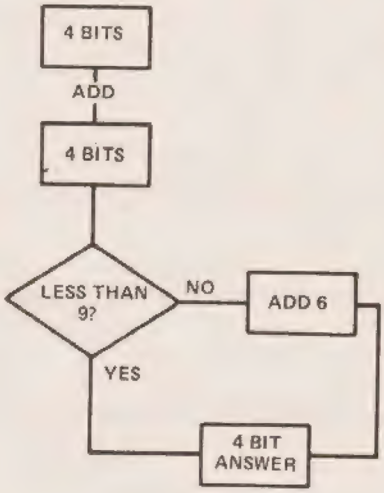
Now try an example which differs from ordinary binary addition. RESET and CANCEL, then load in BCD 39, then add 57 to get 96. Fig. 2 shows this addition, and also shows that the BCD addition gives a very different answer from the same figures added in pure binary fashion. The practical procedure is 11000100 (LDI) PUSH 00111001 (BCD39) PUSH 11101100 (DAI) PUSH 01010111 (BCD57) PUSH 11001000 PUSH 00000001 PUSH to display. The result is 10010110 as expected, BCD 96. The carry/link bit operates on BCD additions also; it is added in the lowest order bits if the logic of the BCD arithmetic requires it, and is set by a carry out of the highest order bit.

So much for addition – what about subtraction? Binary subtraction is achieved by converting the number which is to be subtracted into a negative number, using the 2's complement form as illustrated in Fig. 3. The 2's complement is formed by complementing the number (that is, exchanging a 1 for a 0, or a 0 for a 1 in each bit) and then adding 1 to the least significant bit (lsb). If the binary numbers which are being subtracted consist of more than one byte, however, it is only the lowest order byte which needs to be 2's complemented; the higher order bytes are simply complemented and then added (Fig. 4).

The Byting Order

Because of this distinction between lower and higher order bytes, which can be made only by the programmer, the 8060 instruction COMPLEMENT AND ADD IMMEDIATE (CAI) is a simple complement, not a 2's complement, followed by binary addition. To form a 2's complement, an additional 1 has to be added in, and this is most easily done by making use of the CY/L bit, since this is automatically added in when the CAI instruction is used.

The subtraction routine therefore starts, after the usual RESET and CANCEL, with SET CY/L (SCL), coded as 00000011, PUSH. We can then load a number decimal 103 in this example by using the steps 11000100 PUSH 01100111 PUSH. The number to be subtracted is 32, and the steps are 111111100 (CAI) PUSH 00100000 (data 32) PUSH. The results are displayed as usual by 11001000 PUSH 00000001 PUSH, giving the result 01000111, which is




```
EXAMPLES:  23              0010    0011
         + 42              0100    0010
         ----              ------------
         = 65              0110    0101
                      less than 9  less than 9

           76              0111    0110
         + 48              0100    1000
         ----                      ----
         =124                      1110     MORE THAN 9
                                   0110     ADD 6
                                   ----
                                   0100
                                   ----
                                  (carry)

                           ----
                           1100    MORE THAN 9
                    1      0110    ADD 6
                           ----
                 CARRY     0010
             ---------------------------
             0001    0010      0100
             ---------------------------
               1       2         4
```

Fig.1. The rules of BCD addition. Each group of four bits is binary added, including any carry from a previous stage. If the sum of four bits is less than 9, the result is valid. If the sum is more than nine, 6 is added.

decimal 71, the correct answer. This chip knows a thing or six, doesn't it? Now find out for yourself whether the carry of overflow bits are set, using the CAS and display steps as before.

The subtraction which we've illustrated is a straightforward one, with a positive answer, indicated by a zero in the most significant place. Now try a subtraction which has a negative answer, as shown in Fig. 5. We start again with the SET CY/L instruction, and load the binary equivalent of 69. This is done using 00000011 PUSH 11000100 PUSH 01000101 PUSH. Now follow with 11111100 (CAI) PUSH 01111000 (decimal 120) PUSH, and display by using 11001000 PUSH 00000001 PUSH. The result which is displayed is 11001101, the result of straightforward binary arithmetic. We want to interpret this as a signed number, and it'll be negative because of the 1 in the msb. To find the equivalent negative decimal number, we must subtract 1 and complement, the opposite of the 2's complement procedure, giving 00110011, which is decimal -51. Can we program the microprocessor to get this binary number by itself?

Complement Yourself

Complementing a binary number in 2's complement form is a bit of a roundabout procedure, but here goes. First of all, we have to subtract 1 from the original result, 11001101. This is itself done by a 2's complement method, using the CAI instruction followed by a string of eight zeros. These zeros will complement to 11111111, which is the 2's complement form of -1. The steps are 11111100 PUSH 00000000 PUSH. Now how do we complement the number which is stored in the accumulator? One way to do this on the 8060 is by using the extension register, a very useful feature of this chip. The extension is another single-byte register, which has a similar set of instructions to these which act on the accumulator. The instruction XAE exchanges the byte in the extension register with the byte in the accumulator, so that the data byte 11001100, formed by the subtraction, is now in the extension. If the chip was previously reset, there will be nothing in the extension register, so that the accumulator will now be filled with zeros. If this little lot were being carried out in the middle of a large program, we would probably want to clear the extension register beforehand by loading zeros into the accumulator and exchanging before the arithmetic steps.

The next instruction is CAE, which complements the number in the extension register and then adds it into the accumulator. Any carry bit will also be added in, so we must be certain that there is no carry bit set before we carry out this instruction. Assuming that the carry was not reset earlier, the sequence is now 00000010 (CCL) PUSH 00000001 (XAE) PUSH 01111000 (CAE) PUSH followed by the usual procedure for displaying the result. The whole set of steps is shown in order in Fig. 6 just to remind you of what we've done. The answer should be the binary equivalent of the decimal number. When would you go through this? When the msb is 1 after a subtraction, and when the state of the OV bit indicates that it's correct to take the msb as a genuine sign bit — could you write these instructions as a program?

Changing Pans

So far, we've been using the program counter simply as a counter, going up one count at each program step. The most interesting and useful program steps, however, involve jumping from one program number step to another which is NOT in sequence. This means that the numbers on the address lines will also be out of sequence; remember part of the memory. Instructions which result in out-of-sequence moves like this are transfer instructions, and are transfer instructions, and each microprocessor chip has a variety of types of these transfer instructions. In fact, the most marked differences between different types of microprocessors lie in the varieties of transfer instructions which they can use.

The simplest and most straightforward method of transfer is called program-relative displacement. To see how it works, we'll use the instruction JMP, which simply means JUMP TO A NEW ADDRESS; its instruction code is 10010000 on the INS8060. Start by the usual RESET and CANCEL, then set up the JMP code. This is the instruction part of a two byte code, which must be followed by a data byte, so that when PUSH is pressed, the address count goes from 0001 to 0010 and waits. Now enter the data byte 00000011, which is the binary form of decimal 3. You would expect on previous form that the next PUSH would enter this data and take the address to 0011 — try it.

Now what has happened is that the data byte which you entered just after the instruction byte has been added to the address number, it hasn't gone into the accumulator. What would go into the accumulator, if we had a fully-fledged system operating, would be the number which is stored in the memory at this new address; in this example address 0110.

That looks fine if we want to make the address shift forward — but by how much? Try again, resetting and then entering 10010000 PUSH again. Now enter the data byte 11111111 PUSH. The address might be expected to move to 0011 — does it?

The entry of 11111111 has behaved like a subtraction of one, which means that the leading 1 of the data byte has been interpreted by the microprocessor as a negative sign. The byte of 1's is, in fact, the 2's complement form of -1, so that the address can be moved backwards as well as forwards, as we might expect.

Just to be sure, enter 10010000 again, then PUSH

```
          00111001
BCD Add   01010111
          --------
          10010110
```

Fig.2. A BCD addition of two digits each byte.

```
10011011  NUMBER
--------
01100100  1's COMPLEMENT
      +1
--------
01100101  2's COMPLEMENT
```

Fig.3. Forming the 2's complement for a subtraction.

```
                       upper byte    lower byte
                        11001111      10100101
                      - 01100110      11000011
                        ----------------------
   is equivalent to :

                        11001111      10100101
   + 1's complement -   10011001      00111101   - 2's complement
            1           01101000      11100010
          carry         ----------------------
```

Fig.4. A sixteen-bit number is 2's complemented by the usual method of forming the ones complement and then adding 1. This is the same as 2's complementing the lower byte and ones complementing the higher byte.

Subtraction	01000101	69
−	01111000	−120
	11001101	− 51

Converting answer out of 2's complement form:

11001101	
−1	subtract 1
11001100	
00110011	complement = −51

Fig.5. A subtraction which gives a negative answer, and the method of finding the size of the answer by inverting the twos complement.

Mnemomics	Description
CAI	Complement and add 0
0	to the number in the accumulator
CCL	clear carry/link
XAE	exchange accumulator with extension register
CAE	complement extension and add to accumulator
DISPLAY	and display

Fig.6. The program steps for inverting a 2's complement.

and follow with 11111110, which is -2 in 2's complement form, and PUSH. Once again, the expected address would be 0011 – what is it?

Jump For Joy?

The JMP operation is a clean break in address counting, and the count will then go on from the new address. Try these examples again, watching the address LEDs carefully, and following the data byte with the new instruction 00001000 (NOP), PUSH as many times as you like. This instruction is NO OPERATION, and its use lets you concentrate on what the address LEDs are doing without any data bytes needing to be entered.

Let's stop for breath. Program-relative displacement lets you shift the address, using a byte following the instruction byte as a 'displacement' number – a number which specifies how many places forward or back the address must shift. Because this byte is interpreted by the microprocessor itself as a signed number, the largest positive value must be 01111111 (which is decimal +127), and the largest negative number must be 10000000 (which is decimal – 128). This limits the range of addresses we can get to, but it's perfectly adequate for most purposes, particularly the purposes for which a Simple Cost-effective MicroProcessor might be used. The reason for using this method of transfer is straightforward – an address consists of two bytes, and if we use only one byte of address we can only change the lower byte. Some other microprocessors have a memory addressing system in which both bytes of an address can be loaded in directly, one at a time, so that there are three-byte instructions. We'll look at a less-direct variation, which allows a full address number to be loaded, in the next part.

Before we leave the JMP instruction, though, there are a few points which are worth noting. One is that the basic instruction of JMP is 100100, and the remaining two bits specify which register is used as the 'base' or 'index' address, to which the displacement number is added. We've used 00 for these last two bits of the instruction, making the jump relative to the count number on the program-counter register – more of this later. The other point is that conditional jumps are possible. The code 100101 is the start of the JP instruction, meaning jump if positive or zero. Here again, completing the byte with 00 causes the jump to be relative to the program counter. If we use 10010100 as a jump instruction there will be no jump if the accumulator contains a negative number – that is if the most significant bit in the accumulator has the value 1. Try it – load in –127 by using the sequence (after resetting) of 11000100 PUSH 10000000 PUSH. Then use JP and watch the address LEDs carefully. With the JP instruction loaded by 10010100 PUSH, the address will now be 0100. Make the jump displacement three places by entering 00000011 PUSH. What address have we reached? Has there been a jump?

Now try again, but enter the data byte 00000001 (which is +1) instead of –127. Is there a jump this time?

There are two other lots of jump instructions. The jump-if-zero (JZ) instruction is basically 100110; if we are displacing relative to the program counter this becomes 10011000. The jump-if-not-zero (JNZ) is basically 100111 (10011100 for program-relative displacement). Try them out, with the accumulator loaded first of all with zero, then with a positive number and then with a negative number.

Immediately Relative Programming

Before we leave program-relative address transfers for the moment, it's important to note that all of the instructions which we have used as IMMEDIATE instructions, such as LDI, ANI, ORI, XRI, DAI, ADI, CAI, have their counterparts as memory reference instructions. This means that the instruction code can be followed by a byte, the displacement byte, which causes the shift of address we have noted for the jump instruction. We've used this, in fact, as a method of idsplaying the quantity in the accumulator, by keying in the instruction STORE (11000100) followed by a displacement of 1 (00000001).

Let's now look at this procedure a bit more closely, watching what happens to the address LEDs rather than the data LEDs. Reset, and load up 11001000, the ST instruction. The address will be 0001 until PUSH is operated, when the address changes to 0010. Now set up the displacement number decimal 3, as 00000011 and PUSH. The next address would normally be 0011, but is in fact 0101, three steps on from 0010. Note that it's NOT three steps on from 0011. Now set up 00001000, the NOP instruction, and PUSH again. What happens to the address number now?

Unlike the JMP 3 instruction, in which the address number moves three places on from the program count, and then carries on counting from there the LOAD instruction, followed by a displacement of 3, causes the address of the first part of the instruction to jump three places, but then returns to the normal count at the next PUSH. Look at this one again, comparing JP3 (11001000 PUSH 00000011 PUSH) with LD3 (11001000 PUSH 00000011 PUSH). Another way of looking at the effect of the displacement is that the program counter is incremented by 1 in addition to the displacement on each JUMP instruction, but this increment does not take place on any of the other memory transfer instructions.

In fact that's not quite true, because we can specify what's called auto-indexing on some instruction, but we're not quite at that stage yet. Next month, another LED is added, and we take the plunge into the pointer registers of the INS8060.

8 BITS OR 16 BITS

Do you want a microcomputer kit based on a 16 bit microprocessor? Or an 8-bit microprocessor? With a microassembler? Supply your own ASCII encoded keyboard? No need to look further as BL Microelectronics has developed the BIPROC microcomputer kit with choice of two microprocessors on a common PTH board.

MICROPROCESSORS

* 16 bits TMS 9980-69 instructions including multiply and divide — 7 addressing modes including indirect and memory to memory — 16 general purpose registers file positioned anywhere in memory.
* 8 bits Z-80 — firmly established in the hobby and academic fields

BOARD FEATURES

* Monitor and microasembler resident in EPROM
* 1K User RAM, upto 4K EPROM
* TV, cassette and RS232C memory-mapped interface
* 16 lines of 64 characters each on TV screen
* Programmable character generator
* Interface for ASCII encoded keyboard
* Expandable. * Manual-some useful programmes listed

PRICE

* BIPROC kit 1	Z-80 only	**£164** + VAT
* BIPROC kit 2	TMS 9980 only	**£169** + VAT
* BIPROC kit 3	Z-80 and TMS 9980	**£195** + VAT
	Keyboard not included in above kits	
* 57 pos. keyboard suitable for above kits		**£30** + VAT
* Expansion board for upto 4K RAM (2114s)		**£8** + VAT

Details and demonstration at

WATFORD ELECTRONICS, 33/35 Cardiff Road, Watford, Herts. WD1 8ED
Tel. Watford 40588/9

Trade only B. L. MICROELECTRONICS, 1 Willow Way, Loudwater, Bucks. HP111JR

You've heard about it — read about it — here it is

Available EX-STOCK complete kit as per manufacturers spcification
with provision for 8k on board expansion excludes 4118 x 8 *

SCOOP OFFER
NASCOM — 2
ONLY 500 KITS AVAILABLE at this price

INCLUDES **FREE** 16K EXPANSION
Value £140 includes **ALL** parts with every kit

NASCOM — 2
on demonstration now

£295 + 15% VAT with FREE 16k Expansion worth £140

Available only from us on the coupon below

OPTIONAL EXTRAS
3 amp power supply. **£29.50** VAT 15%, post £1.50 for Nascom — 2

RS232 compatible — **80 COLUMN PRINTER,** list price £550, brand new, if sent by carrier £5 extra. Our Price **£325** + 15% VAT

8 OFF 4118 * for Nascom — 2 purchasers **£80** early delivery

Full manufacturers warranty — don't delay order today

Please send me my NASCOM — 2 kit with FREEK 16K EXPANSION For £295 + VAT. I enclose remittance. . . . . to cover

Name and Address . . . .

. . . .

. . . .

Also in stock Nascom — 1 • Elf • TRS 80 as previously advertised

HENRY'S
COMPUTER KIT DIVISION
404 EDGWARE ROAD,
LONDON W2 ENGLAND
01-402-6822

TRS 80 SOFTWARE

from the leaders in innovative software

We were first for the TRS 80 with Re number Basic

We were first with Level III

We were first with the Howe Monitors 1, 2 and 3.

We were first with Newdos

And we are going to go on being first
SEND SAE FOR LIST

A. J. HARDING
(MOLIMERX)
28 Collington Ave.,
Bexhill, E. Sussex.
Tel. (0424) 220391

BARCLAYCARD VISA

Buy it with Access

the new
TEXTIE MARK FIVE
word processor

For NASCOM 1 with B-BUG or T4 and a Minimum of 8K RAM Expansion at 1000H

!powerful features!

* WORD WRAP-ROUND ON VDU & PRINTER
* UP TO 120 COLUMN LINE LENGTH
* AUTO LATERAL OFFSET MOVES PAGE SIDEWAYS WHEN TYPING LONG LINES
* FULL PRINTER CONTROL margins, line length & spacing, word wrap-round
* POWERFUL EDIT FACILITIES
* SELECTIVE & RELOCATABLE ERROR CHECKED CASSETTE READ AND WRITE
* LATCHABLE SHIFT, CURSOR, ETC.,ETC.

ORDER MARK FIVE
TODAY DIRECTLY FROM
THE SOFTWARE PUBLISHING COMPANY
8A CHURCH SIDE, MANSFIELD, NOTTS
Telephone: (0623) 29237
On Cassette with documentation and listing Mark Five sells at 13.50 including P/P Access Welcome
PLEASE STATE WHICH MONITOR YOU USE

THE NEXT STEP FORWARD IN TIME

Only CASIO technology offers such high quality at such low cost

MICROPROCESSORS

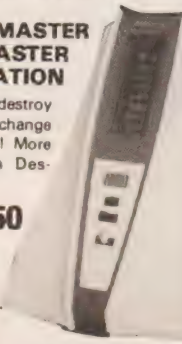
FAIRCHILD GRANDSTAND
Video Entertainment Computer **£74.50**

19 plug-in pre-programmed memory cartridges now available.

Each **£12.50**

Hand held COMPUTER GAMES

We have selected the best available
It's you against the the computer!



UFO MASTER BLASTER STATION

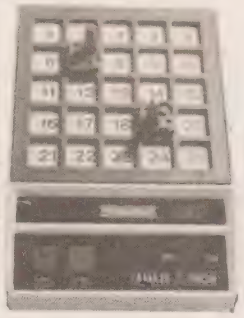
Guide missiles to destroy UFOs which may change course or disappear! More sophisticated than Destroyer.

£22.50

GRANDSTAND 4-in-1

Calculator/Auto Race/Code Breaker (as Digits)/Blackjack. Convenience — Skill — Brainpower — Luck?

£22.50



AMAZE-A-TRON

Over 1 million mazes to defeat! 1 or 2 players. Electronic sound effects.

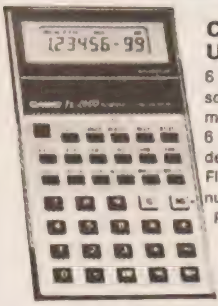
£17.25

GRANDSTAND

Solitair. Absorbing and challenging **£14.95**
Destroyer. Protect your convoy against enemy torpedoes **£14.95**
DIGITS. (As Mastermind). Deduce four random numbers **£13.95**

Most CASIO products available from stock. Send 25p for illustrated brochures and membership of our Christmas club. EXTRA DISCOUNTS on many items.

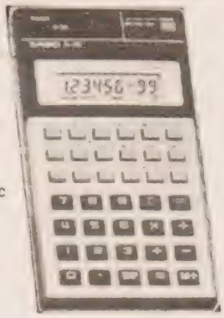
NEW SCIENTIFICS



CASIO FX-2600
Ultra slim (5/32")

6 + 2 digit LC Display. 43 scientific functions. Non volatile memory, auto power off facility, 6 level parentheses, standard deviations, R to P, P to R, ENG, FIX, SCI, RND and random number. 5/32 x 2¾ x 4⅞, wallet

RRP £22.95 **£19.95**



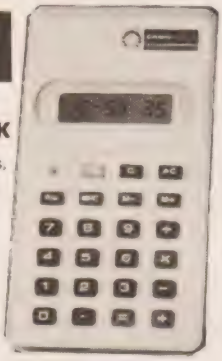
CASIO FX-3200
As above but 10 digits
£21.95

CASIO FX-310
As FX-2600 but 50 scientific functions, ¼ x 2⅞ x 5¼"
RRP £19.95

CASIO FX-510
10 digit version of above
£19.95

NEW FROM CASIO

HQ-21
CALCULATOR/CLOCK



LC display of hours, minutes, seconds.
8 digit calculator with full memory, percentages, etc.
Two AA batteries give very long life
¾ x 3 x 5⅝"
RRP £12.95 **£10.95**

From Casio's New Collection comes one of the most sophisticated executive watches available today.

THE 81-CS-36B ALARM CHRONOGRAPH

LC Display of hours, minutes, seconds, day; And with day, date, month and year perpetual automatic calendar.

4-5 YEAR BATTERY
1/100 second chronograph to 7 hours.
Net. lap and first and 2nd place times.
User optional 12 or 24 hour display.
24 hour alarm.
User optional hourly chime.
Backlight.
Mineral glass.
Stainless steel case.
Water resistant to 100ft (3 at.)
RRP £39.95 **£35.95**

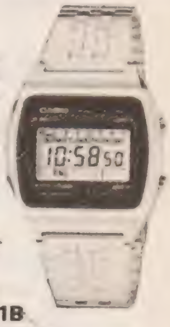
HONGKONG WATCHES

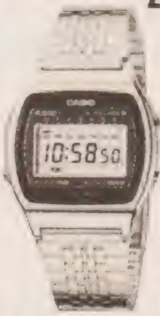
Most low cost watches come from Hongkong. In our experience these are proving to be extremely unreliable, particularly those with multi-function modules, with failure rates of up to 60% or more. Repairs can take as long as three months and replacement parts are not always available.
Compare this with Casio, Citizen and Seiko, whose failure rate is typically under 1% and Casio's service time of 2-3 weeks and we ask you:
ISN'T IT WORTH PAYING A LITTLE MORE FOR QUALITY AND RELIABILITY? Fully guaranteed for 12 months.

CASIO CHRONOGRAPHS

CASIO 95QS-31B

4 YEAR BATTERY
1/100 sec. chrono to 7 hours. Dual time. 12 or 24 hour. Stainless steel encased. Water resistant to 66 feet (2 at.)
RRP £27.95 **£23.95**



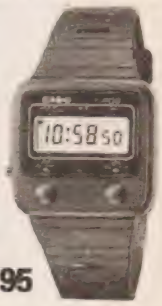


CASIO 95CS-31B

5 YEAR BATTERY.
1/100 sec. chrono to 7 hours. Dual time. 12 or 24 hour. Solid stainless steel case. Water resistant to 100ft (3 at.). RRP £34.95
£29.95

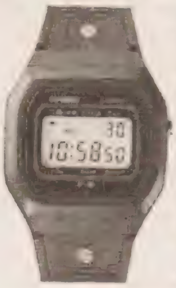
Both have new Lithium batteries which outlast most SOLAR watches. Constant LCD display of hours, minutes, seconds, am/pm and day, (12 or 24 hour). Dual time (12 or 24 hr). Automatic day, date, month and year calendar. Mineral glass face. Backlight. High quality s/s bracelets with easily removable links.

CASIO F-200
Sports chrono



Hours, minutes, seconds, am/pm; and with day, date and month auto calendar.
1/100 sec. chrono to 1 hour.
Net. lap and 1st and 2nd place times.
Resin case and matching strap.
Mineral glass.
Water resistant to 66ft. (2 at).
Silver oxide battery.
RRP £17.95 **£15.95**

8 DIGITS — TIME/DATE



CASIO F-8C
3-Year Battery

8-digit display of hours, minutes seconds and date, with day and am/pm.
Auto. calendar.
Backlight.
Resin case and matching strap.
Mineral glass.
Water resistant to 66ft. (2 at).
RRP £12.95
Real quality and value for money.
£10.95

CASIO LADIES WATCHES

Superbly styled stainless steel cases, with mineral glass face, water resistant to 100 feet. LC display of hours, minutes, 10 seconds, seconds by flash, am/pm; and with day, date and month.

86QL-10B	86QL-12B	86CL-22B
£26.50	**£31.95**	**£37.25**

Price includes VAT and P&P. Send your Company order, cheque, P.O. or phone your ACCESS or BARCLAYCARD number to:—

TEMPUS

Dept CT, Beaumont Centre, 164-167 East Rd., Cambridge CB1 1DB. Tel. 0223 312866

Keith Ridgway

This months feature program gives you a chance to sink the baddies in a computer version of the board game

This program is written for the Research Machines 380Z and plays a simplified version of the well known game battleships. The computer draws some randomly positioned ships on the monitor and the user attempts to sink the ships with torpedoes from his submarine.

The Program Explained

```
100  REM **************************
110  REM *** BATTLESHIPS MK II ***
120  REM **************************
130  REM *** INITIALISATION ***
140  WIDTH 80
150  CLEAR 100
160  GOSUB 1160
170  INPUT"HOW MANY SHIPS"; S
180  IF S>9 OR S<0 THEN ?"INCORRECT (1 - 10)":
     GOTO 170
190  GRAPH 1
200  H=0:G=0
```

The 'width 80' on line 140 is only there to make the format of the instructions easier. Clear 100 reserves some memory for use in string variables. Graph 1 sets most of the screen up as a piece of graph paper. H and G count the hits and the number of goes respectively.

```
210  REM *** SET TITLE ***
220  T$="BATTLESHIPS"
230  FOR I=1 TO LEN(T$)
240       PLOT 31+I*2-2,47,ASC(MID$(T$,I,1))
```

BATTLESHIPS

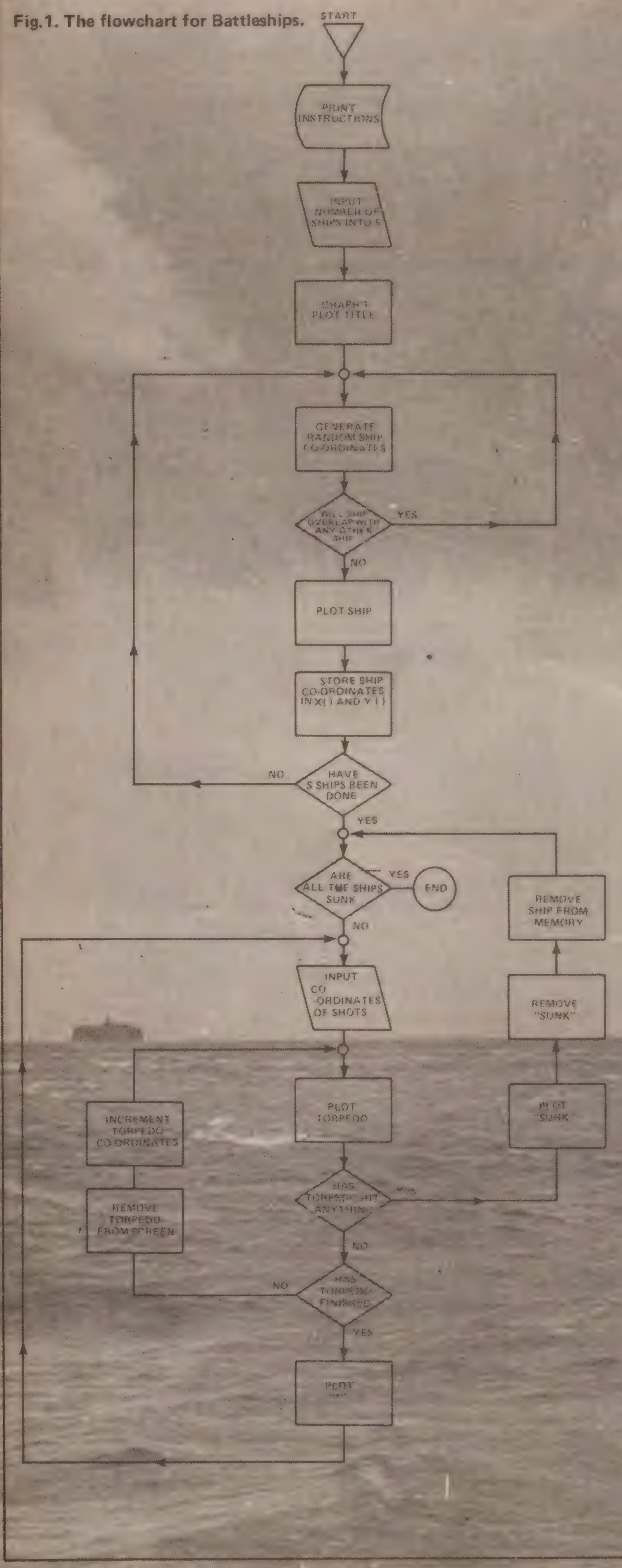
```
250  NEXT
260  FOR I=30 TO 50 STEP 2
270          PLOT I,44,96
280  NEXT I
```

This portion of the program is not essential: it 'plots' the title of the game at the top of the screen. The plot statement is in the form PLOT Xco-ordinate, Yco-ordinate, ASCII code.

```
290  REM  ***  SET TARGET  ***
300  RANDOMIZE
310  FOR I=1 TO S
320          Y=INT(RND(1)*46)
330          Y=3*(INT(Y/3))
340          X=INT(RND(1)*66)+6
350          X=2*(INT(X/2))
360          REM *** CHECK FOR SHIP OVERLAP ***
370          FOR J=1 TO S
380            IF Y<>Y(J) THEN 440
390            IF X=X(J)-8 OR X=X(J)+8 THEN 320
400            IF X=X(J)-6 OR X=X(J)+6 THEN 320
410            IF X=X(J)-4 OR X=X(J)+4 THEN 320
420            IF X=X(J)-2 OR X=X(J)+2 THEN 320
430            IF X=X(J)  THEN 320
440          NEXT J
450          REM  ***  PLOT SHIP  ***
460          PLOT X-4,Y,184
470          PLOT X-2,Y,176
480          PLOT X,Y,189
490          PLOT X+2,Y,184
500          PLOT X+4,Y,144
510          REM  ***  STORE CO-ORDS. OF BRIDGE  ***
520          X(I)=X
530          Y(I)=Y
540  NEXT I
```

Lines 320 – 350 generate co-ordinates for the ship and round them off to make the game run better. Lines 370 – 440 compare these co-ordinates with the co-ordinates of the other ships to ensure that the ships do not overlap. Lines 460 – 500 plot the ship onto the screen and then its co-

Fig.1. The flowchart for Battleships.

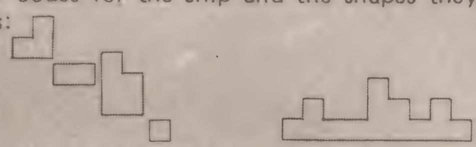


BATTLESHIPS

ordinates are stored in X() and Y(). This part of the program is a loop so that the number of ships input at the beginning are generated on the screen.

The ASCII codes for the ship and the shapes they form are as follows:

184 :
176 :
189 :
144 :



They go together to form a ship which looks like this:—

Shooting And Hitting

```
550  REM *** INPUT GUESS SHOT ***
560  IF H=S OR G=10 THEN 1060
570  ?"YOU HAVE HAD";G;"GOES,AND";H;"HITS"
580  INPUT"WHAT IS X CO-ORDINATE(1–79)";A
590  INPUT"WHAT IS Y CO-ORDINATE(1–46)";B
600  A=2*(INT(A/2))
610  B=3*(INT(B/3))
620  IF A>79 OR B>46 OR A<0 OR B<0 THEN
     ?"CO-ORDS. OUT OF RANGE":GOTO 580
630  G=G+1
```

The co-ordinates for the shot are input at lines 580 and 590: they are rounded off as for the ship co-ordinates. If the co-ordinates are too big or too small an error message is printed at line 620. The 'goes' counter is incremented by one.

```
640  REM *** TORPEDO ***
650  A1=0:B1=0
660  FOR I=1 TO 100:NEXT I
670  PLOT A1,B1,32
680  IF A1<A THEN A1=A1+2
690  IF A1>A THEN A1=A1–2
700  IF B1<B THEN B1=B1+3
710  IF B1>B THEN B1=B1–3
720  PLOT A1,B1,204
```

A1 and B1 hold the torpedo co-ordinates and are incremented or decremented as required by lines 680 – 710 such that the torpedo heads for the shot co-ordinates. ASCII code 204 is a horizontal bar.

```
730  REM *** CHECK FOR HIT ***
740  FOR I=1 TO S
750       X=X(I):Y=Y(I)
760       IF B1=Y THEN 770 ELSE 820
770       IF A1=X–4 THEN 870
780       IF A1=X–2 THEN 870
790       IF A1=X THEN 870
800       IF A1=X+2 THEN 870
810       IF A1=X+4 THEN 870
820  NEXT I
830  IF A1<>A OR B1<>B THEN 660
840  PLOT A,B,42
850  GOTO 570
```

Lines 760 – 810 compare the co-ordinates of the torpedo with all the stored ship co-ordinates in turn. If they agree the computer goes to line 870 for the hit routine, if not the program either loops back to move the torpedo again or it plots a star where the torpedo explodes.

```
860  REM *** HIT ROUTINE ***
870  PLOT A1,B1,42
880  REM *** REMOVE SHIP FROM MEMORY ***
890  X(I)=-1:Y(I)=-1
900  FOR I=1 TO 900:NEXT I
910  REM *** PLOT "SUNK" ACROSS SHIP ***
920  PLOT X-4,Y,83
930  PLOT X-1,Y,85
940  PLOT X,Y,78
950  PLOT X+3,Y,75
960  H=H+1
970  FOR I=1 TO 1000:NEXT I
980  REM *** REMOVE SHIP ***
990  PLOT X-4,Y,32
1000 PLOT X-2,Y,32
1010 PLOT X,Y,32
1020 PLOT X+2,Y,32
1030 PLOT X+4,Y,32
1040 GOTO 550
```

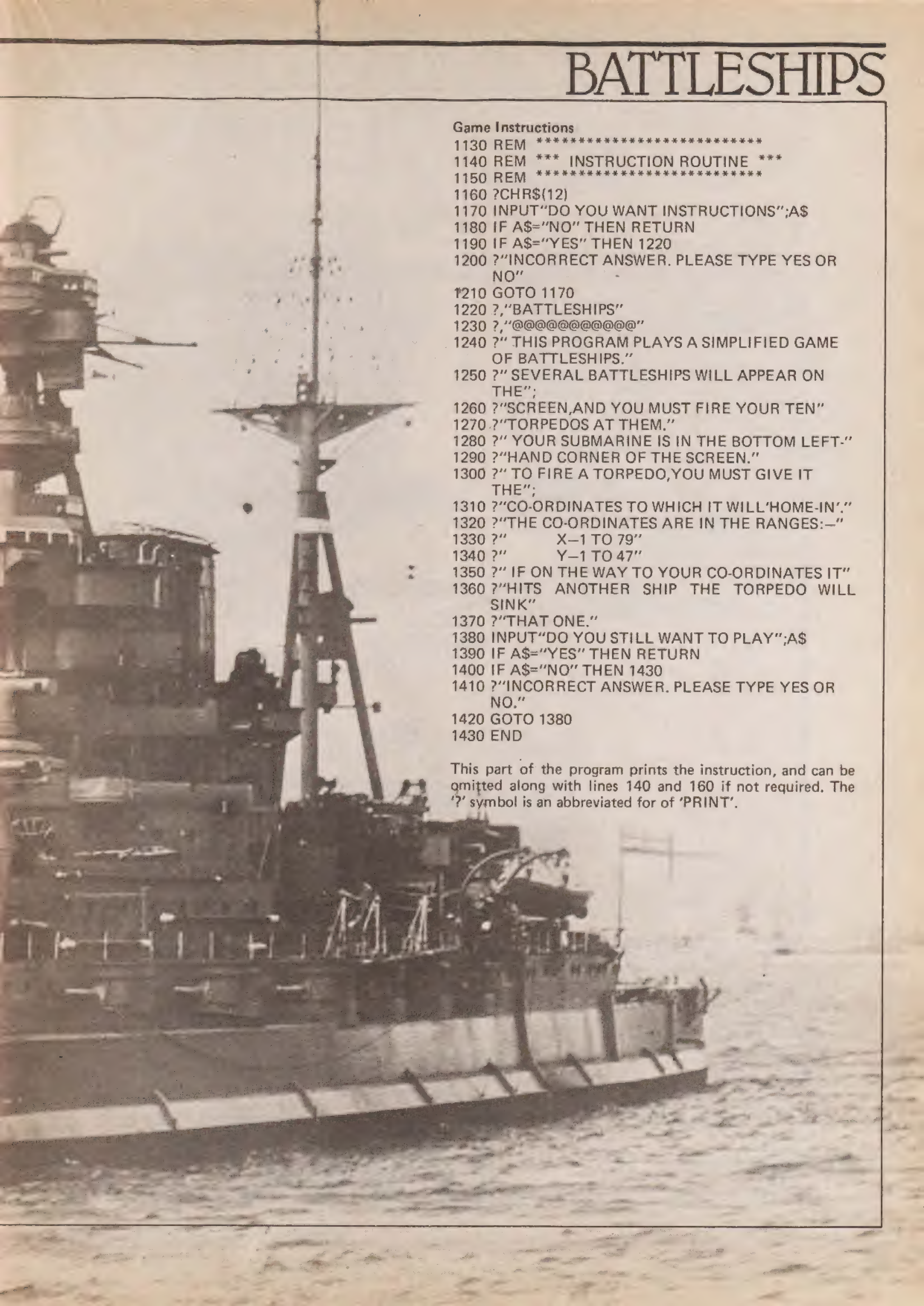
This part of the program moves the ship off the screen, writes 'sunk' across where the ship was, and then clears that part of the screen by plotting spaces over it. At line 960 the 'hits' counter is incremented by one. It then returns to line 550 to input another shot.

The Endgame

```
1050 REM *** GAME OVER ***
1060 GRAPH 0:? CHR$(12)
1070 ?"YOU HAVE HAD ";G;" GOES,AND ";H;"HITS"
1080 INPUT"DO YOU WANT ANOTHER GAME";A$
1090 IF A$="YES" THEN 110
1100 IF A$="NO" THEN 1430
1110 ?"INCORRECT ANSWER. TYPE YES OR NO."
1120 GOTO 1080
```

The computer only reaches here when all the ships are sunk. Graph 0 resets the screen from Graph 1. The command "?CHR$(12)" clears the screen and resets the cursor.

BATTLESHIPS



Game Instructions

```
1130 REM ****************************
1140 REM *** INSTRUCTION ROUTINE ***
1150 REM ****************************
1160 ?CHR$(12)
1170 INPUT"DO YOU WANT INSTRUCTIONS";A$
1180 IF A$="NO" THEN RETURN
1190 IF A$="YES" THEN 1220
1200 ?"INCORRECT ANSWER. PLEASE TYPE YES OR
     NO"
1210 GOTO 1170
1220 ?,"BATTLESHIPS"
1230 ?,"@@@@@@@@@@@"
1240 ?" THIS PROGRAM PLAYS A SIMPLIFIED GAME
     OF BATTLESHIPS."
1250 ?" SEVERAL BATTLESHIPS WILL APPEAR ON
     THE";
1260 ?"SCREEN,AND YOU MUST FIRE YOUR TEN"
1270 ?"TORPEDOS AT THEM."
1280 ?" YOUR SUBMARINE IS IN THE BOTTOM LEFT-"
1290 ?"HAND CORNER OF THE SCREEN."
1300 ?" TO FIRE A TORPEDO,YOU MUST GIVE IT
     THE";
1310 ?"CO-ORDINATES TO WHICH IT WILL'HOME-IN'."
1320 ?"THE CO-ORDINATES ARE IN THE RANGES:-"
1330 ?"          X-1 TO 79"
1340 ?"          Y-1 TO 47"
1350 ?" IF ON THE WAY TO YOUR CO-ORDINATES IT"
1360 ?"HITS ANOTHER SHIP THE TORPEDO WILL
     SINK"
1370 ?"THAT ONE."
1380 INPUT"DO YOU STILL WANT TO PLAY";A$
1390 IF A$="YES" THEN RETURN
1400 IF A$="NO" THEN 1430
1410 ?"INCORRECT ANSWER. PLEASE TYPE YES OR
     NO."
1420 GOTO 1380
1430 END
```

This part of the program prints the instruction, and can be omitted along with lines 140 and 160 if not required. The '?' symbol is an abbreviated for of 'PRINT'.

CT MARKETPLACE

LCD CHRONO

We feel we've got to tell you carefully about this offer which we're introducing for the first time. Why? Because our price is so enormously lower than anywhere else you may suspect the quality.

The display is LCD and shows the seconds as well as the hours — and minutes — press a button and you'll get the date and the day of the week.

Press another button for a coujple of seconds and you have a highly accurate stopwatch with hundredths of a second displayed and giving the time up to an hour. There is a lap time facility as well — and of course a back light.

Our Chrono comes complete with a high grade adjustable metal strap and is fully guaranteed.

£11.95

(Inclusive of 15% VAT and Postage).

To:
LCD Watch Offer
CT Magazine
145 Charing Cross Road,
London WC2H 0EE

Please find enclosed my cheque/PO for £11.95 (payable to CT Magazine) for my LCD Chronograph.

Name ....

Address ....

....

....

Please allow 28 days for delivery

ALARM-CHRONO LCD

This new addition to our unbeatable selection of bargains is no ordinary LCD watch. It's a slim, multi-function, dual time chronograph alarm watch, no less.

This model will show hours, minutes, seconds, date, day of the week, stop watch, split time, alarm and alternate dual time zone — not all at once, of course. There is also a night light.

Hours, minutes, seconds and day of the week are displayed continuously, while the date will appear at the touch of a button. The day of the week is indicated by a flag. When used as a stopwatch, the maximum count is 0.1secs. short of thirteen hours.

£19.95

(Inclusive of 15% VAT and Postage).

To:
ALARM/CHRONO LCD WATCH Offer,
CT Magazine,
145 Charing Cross Road,
London WC2H 0EE

Please find enclosed my cheque/PO for £19.95 (payable to CT Magazine) for my Alarm/Chrono LCD watch.

Name ....

Address ....

....

....

Please allow up to 28 days for delivery.

Dear Sir,

Having entered your "Brands" program on my Pet over the weekend I found several errors which seem to spoil this otherwise superb listing.

The first group of alterations seem to be common to all Basics.

```
  5 DIM A$(14)
160 DIM S(14)
180 DATA 160, 110, 51, 110, 84, 42, 75, 120, 150, 60,
    120,98,82,160
230 FOR I=1 TO 14
450 PRINT "THE SAME THING HAPPENS THE FIRST
    TIME YOU EXCEED 12,000 RPM"
460 FOR MN=1 TO 10,000:NEXT MN
575 M=S(I)*((10+T5)/10)
580 PRINT A$(I)
675 IF G > 5 OR G < 1 THEN 670
690 IF B > 9 OR B < 0 THEN 680
720 IF T > 9 OR T < 0 THEN 710
740 IF B > 8 THEN IF T5 > .87 THEN 1500
905 IF R > 12000 OR Z=6 THEN 1000
```

The following alterations are for PET which has a peculiar random number generator and no "DIGITS" function.

```
264 deleted
570 T5=RND(TI)
600 PRINT "MAXIMUM SPEED ON THIS SECTION IS";
    INT(M);"MPH"
840 PRINT G; TAB(9);INT(V);TAB(17);INT(R);TAB(28);
    INT(Q5)
1300 PRINT "YOU HAVE CRASHED AT";INT(V2+2.2);
    "MPH"
2000 PRINT "YOU COMPLETED THE COURSE IN";
    INT(Q5);" SECONDS WELL DONE!!"
2020 PRINT "YOUR AVERAGE SPEED WAS";
    INT(84*60/Q5);"MPH"
```

These alterations make "Brands" a very enjoyable program to run. Now there is no need to write a rule "Concord Prohibited" as the maximum speeds (line 575) have been reduced to more earthly figures.

May I compliment you on the high run value offered by this program. Will you please advise me if you intend to publish an article on 6502 Machine Code programming.

Yours faithfully,
A.K.Jowett (15).

197 Victoria Road East,
Thornton-Cleveleys,
Blackpool,
FY5 3ST.

Dear Sir,

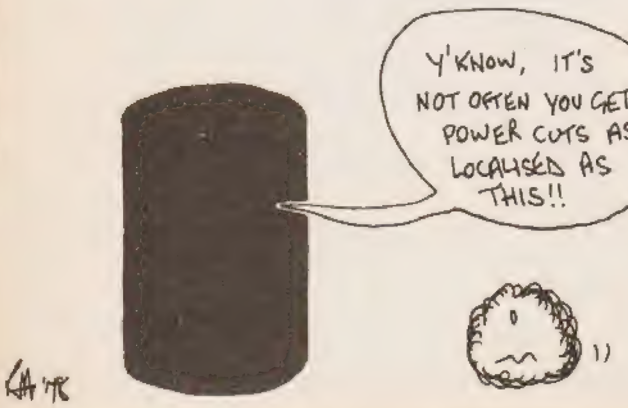
I would be very grateful if you could print a few lines in your Printout column to the effect that I intend to set-up a program exchange. Entitled Micro News, this would be a newsletter (of approx. 10 pages) covering programs for less publisized micros such as the Superboard, Nascom 1 & Exidy Sorcerer.

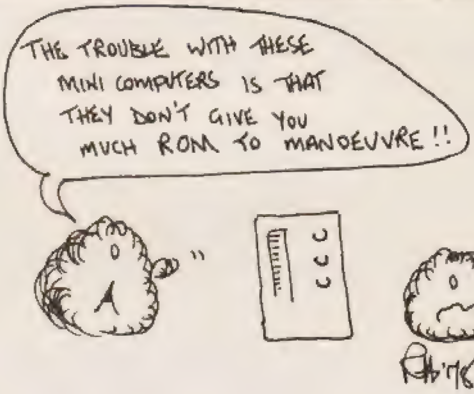
Depending on the number of enquiries, a small charge will be made. This would cover both the cost of printing, and also any postal charges. If anyone is interested, please write direct to me (Martin Black) at the given address. In this connection, an SAE would be greatly appreciated.

Thankyou for a most informative magazine.

Yours sincerely,
Martin Black.

11 Moorland Avenue,
Crumpsall,
Manchester 8,
M8 6WT.





Dear Sir,

I wonder if you are familiar with the subject matter of the enclosed newspaper cutting (Observer Sunday 26th Aug). Why should owners of small computers, whether personal or commercial, pay levies to music – record companies, merely in order to record computer programs? I am sure that there are many other uses for tape – cassettes (apart from music), for example, use in the learning of a foreign language. Perhaps you can make representations to the appropriate authorities?

Yours truly,
J.R.Camp.

48, Eastdean Avenue,
Epsom,
Surrey.
KT18 7SN

Ed: We have spoken to the BPI and they assure us that steps will be taken to prevent the levy being charged on tapes used for "legitimate" purposes. The levy is only a proposal, nothing is definite yet.

Dear Sir,

TRITON V5.1 BASIC

I must write to congratulate you on getting my name on one of my articles at last. Unfortunately because of the number of printing mistakes in it, I would have preferred to have my name omitted.

For those who want to read it, I suggest they take the following route. Start at the beginning on page 14 and read to the end of page 15 making the following corrections.
At the end of page 14 the line should read:
Now type in 123456 Return following by LIST
and the line 130 of the example on page 15 should read:
130 PRINT '   ',

Now the difficult bit. The example at the end of page 15 is continued at the end of page 16 and the article continues from here onto page 17 and down to the heading Intercept Jumps. From here it returns to the top of page 16 and continues over the remainder of this page. It then jumps to the last section on page 17.

The following corrections should be made while reading this part of the article.
Line 40 of the example on page 16 should read:
40 IF B=0 PRINT #1,A, 'IS DIVISIBLE BY',I
and the end of the WRITE description should read:
The data is also the least significant byte of a numeric constant, a variable or an expression.

Yours faithfully,
D.M. Scales.

43, Knighton Road,
Otford,
Kent. TN14 SLD

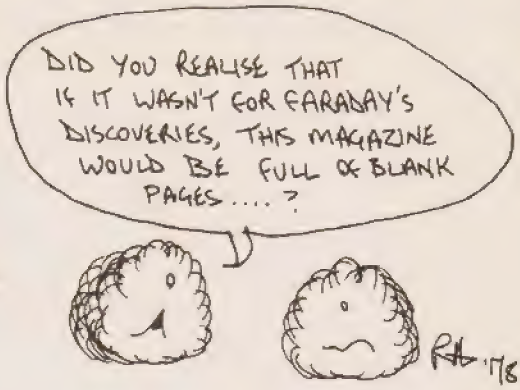
Dear Sir,

Not knowing the 'PET' BASIC at all, I do not know whether it supports the 'ON . . . .GO TO' function. Assuming it does (or for those planning to run the program on BASICs which do) the program 'Hangman' by M.J. Coates (CT, Sept, P.48) would surely run more efficiently were lines 480 to 730 replaced by this:

```
475 on (ASC (Y$) - 17) * 10) GOTO 480, 490, 500, . . .730
480 M = 0
490 M = 2
500
_
_
_
730 M = 104
```

Yours sincerely,
Nick Beard.

High Street,
Braithwell,
Rotherham,
South Yorkshire.



Dear Sir,

Mr A Stephenson's article "APPRECIATING MPUs" CT July 79, is one of the best ones on the subject, at this level, that I have read.

The following are some points which may be relevant to the scope of his article, including some which I originally found not immediately obvious.

P64. Memories
Battery standby can be used to retain programs in otherwise volatile semi-conductor memories.

Bi-directional Data Bus
Earlier busses such as the S100 are not Bi-directional but require a second set of 8 lines for READING.
Also some versions use positive logic (a digital '1' is + 5V) and others may use negative logic, (a digital '1' is + 0V) for the various bus signals.

RESET
End of paragraph ––– a special address "provided for program restart uses". Reset is also used by the interface circuits, to clear-out any registers that may contain incorrect data, perhaps left from a previous interrupted sequence.

INTERRUPT REQUEST
End of paragraph ––– (Which is programmable "to provide means by which a maskable interrupt can be 'held off' under program control).

P66. CHIP SELECTS
The Address line decoding provided in each chip is restricted to the number necessary to access the functions of that chip only (ie. for 10 lines). To selectively access other chips in the system it is necessary to provide external decoding of the other 6 lines of the Address Bus. This is then used to Enable the CHIP SELECT of the particular chip to provide virtually an ON/OFF switch action for each chip.

P70. FETCH AND EXECUTE PHASES
It is impossible to differentiate between successive FETCHED Bytes as to which are the instructions and which are operands. However as the CPU 'Unwinds' them strictly in the sequence that the ASSEMBLER originally generated them, a Data operand should never get acted on as an Instruction. Also, as a general point, LSI peripheral chips (PIAs and so on, which can cost almost as much as the CPU chip) are not mandatory for use with any CPU chip.

For many straightforward purposes, discrete TTL or CMOS is adequate for the necessary address decoding and buffering circuitry and can be simpler to use and program.

Yours sincerely,
G.R.Barry.

25 Pelican Road,
Pamber Heath,
Hants.

Brian Windsley

Primes and Factors

```
000  42  STO
001  02   02
002  05   5
003  42  STO
004  00   00
005  73  RC*
006  00   00
007  32  X:T
008  43  RCL
009  02   02
010  34  √X
011  22  INV
012  77   GE
013  00   00
014  33   33
015  53   (
016  43  RCL
017  02   02
018  55   ÷
019  00   0
020  32  X:T
021  54   )
022  22  INV
023  59  INT
024  67   EQ
025  00   00
026  32   32
027  69  OP
028  20   20
029  61  GTO
030  00   00
031  05   05
032  22  INV
033  86  STF
034  00   00
035  92  RTN
036  43  RCL
037  02   02
038  99  PRT
039  22  INV
040  86  STF
041  01   01
042  92  RTN
043  76  LBL
044  12   B
045  86  STF
046  01   01
047  76  LBL
048  13   C
049  71  SBR
050  00   00
051  00   00
052  87  IFF
053  00   00
054  00   00
055  36   36
056  73  RC*
057  00   00
058  66  PAU
059  87  IFF
060  01   01
061  00   00
062  73   73
063  99  PRT
064  22  INV
065  49  PRD
066  02   02
067  71  SBR
068  00   00
069  05   05
070  61  GTO
071  00   00
072  52   52
073  02   2
074  44  SUM
075  02   02
076  71  SBR
077  00   00
078  02   02
079  61  GTO
080  00   00
081  52   52
082  76  LBL
083  15   E
084  68  NOP
085  01   1
086  42  STO
087  04   04
088  02   2
089  42  STO
090  05   05
091  03   3
092  42  STO
093  06   06
094  06   6
095  42  STO
096  01   01
097  05   5
098  71  SBR
099  00   00
100  00   00
101  22  INV
102  87  IFF
103  00   00
104  01   01
105  13   13
106  69  OP
107  21   21
108  43  RCL
109  02   02
110  66  PAU
111  72  ST*
112  01   01
113  43  RCL
114  03   03
115  32  X:T
116  43  RCL
117  01   01
118  77   GE
119  00   00
120  36   36
121  02   2
122  44  SUM
123  02   02
124  71  SBR
125  00   00
126  02   02
127  61  GTO
128  01   01
129  01   01
130  76  LBL
131  11   A
132  43  RCL
133  03   03
134  32  X:T
135  04   4
136  42  STO
137  01   01
138  73  RC*
139  01   01
140  66  PAU
141  43  RCL
142  01   01
143  22  INV
144  77   GE
145  01   01
146  48   48
147  92  RTN
148  69  OP
149  21   21
150  61  GTO
151  01   01
152  38   38
153
```

The primes and factors program listing. Don't forget to load STO 03.

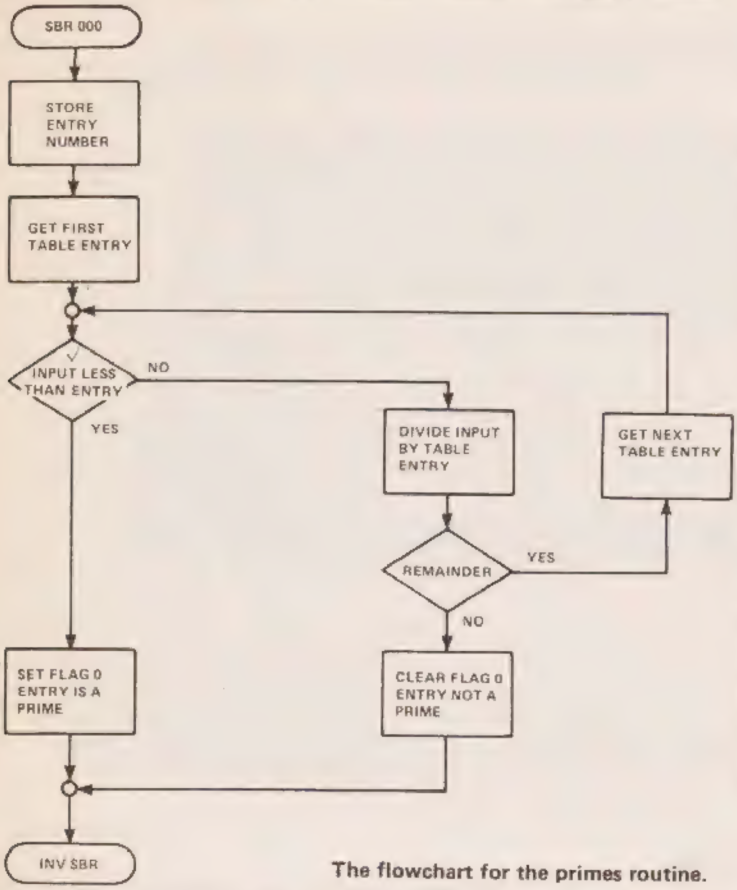
The TI 59 programmable calculator can be used to study numbers up to 249001 (22201 for the TI 58) using available memory capacity. Initially the program at Label E calculates and stores the first 96 primes. These can then be used to factorise an input entry up to the limiting number, or to determine the intermediate primes. The initial store procedeure takes about 25 minutes and the program at Label A can be used to view the table contents.

The flowchart illustrates the mathematics involved, and since there is only one even prime (2) the "next" prime entry at Label B requires an odd number entry to look for an answer.

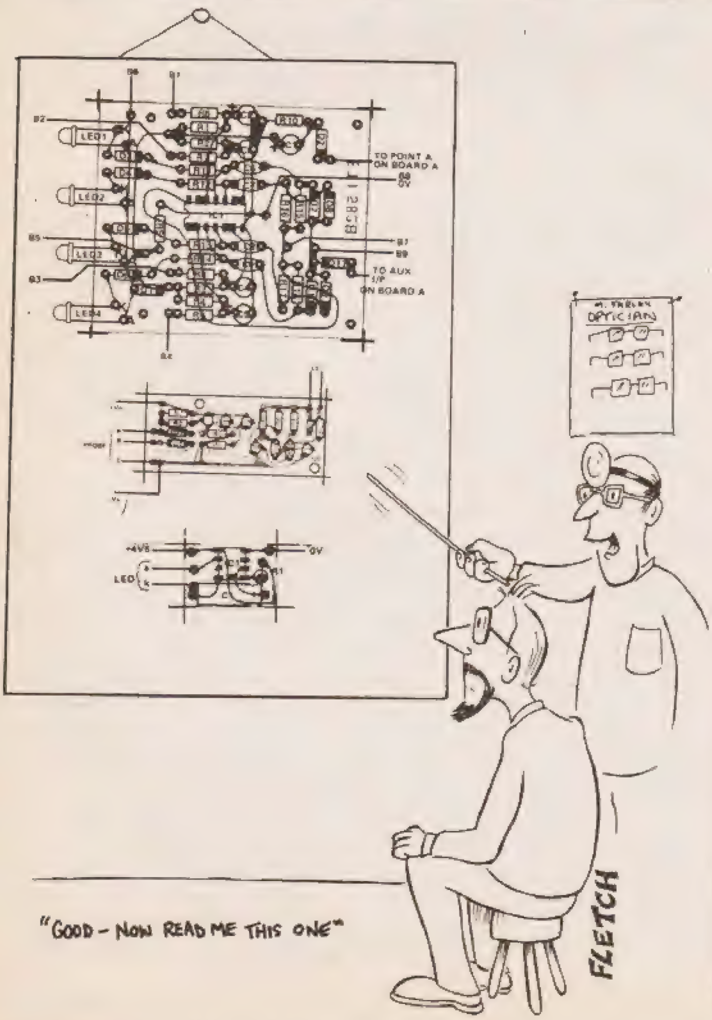
Obviously the magnetic card facility of the TI 59 and the print facility of the PC100B can be used to advantage but the same program using only 40 stores and the TI 58 should prove an interesting diversion.

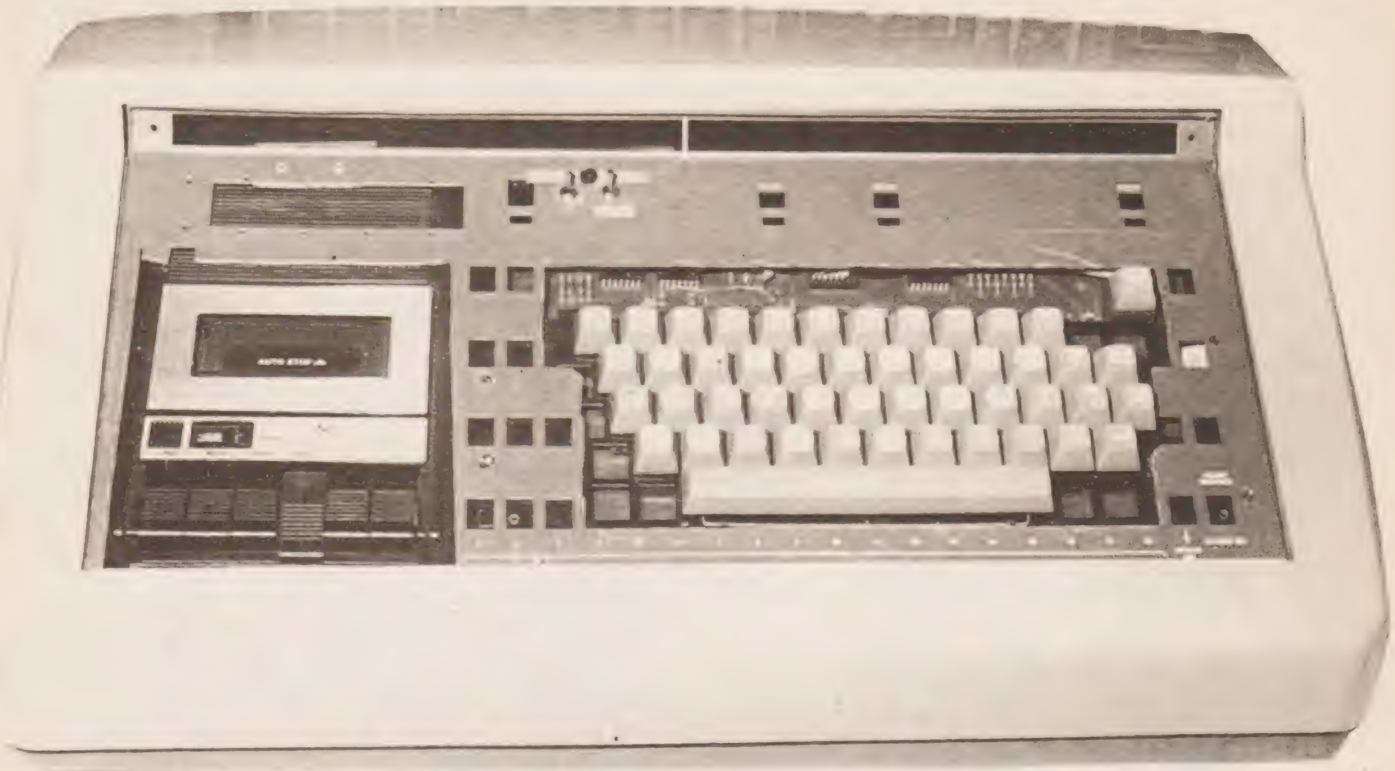
Program Use

Enter the desired table length (max 99 for TI 59, 39 for TI 58) in STO 3. Press E, program takes 25 minutes to run. Now press A to display the table, B to find the next prime from an entered odd number or C to foctorise an entered number. The program includes print commands for use with the PC100B print cradle.



The flowchart for the primes routine.





If you want to perform complex arithmetical operations with your micro this project will make life easy

Over the last two decades we have witnessed a revolution in number processing which has been unsurpassed since the invention of the abacus. Gone is the abacus, gone is the slide rule, even our junior school children are demanding electronic aids that do more than 4 functions at 7 digit accuracy. Now we have the all singing, all dancing microprocessor based home computer, heralded as the salvation to all our domestic problems. The anomaly is the number processing capability of these machines, ranging from very poor to average. If the arguments are in favour of calculators with full scientific or financial facilities encompassing a range of numbers from 1E–99 to 1E99 then those arguments hold good for computers also.

The number handling facilities have been badly neglected by 'chip' manufacturers and software houses have been left to fill the gap. Tiny BASIC gives a range of interger –32767 to 32767 and Extended BASIC's are limited to typic ally 6 digit mantissa E–38 to E38 and are very limited on arithmetic functions.

To obtain the greater accuracy and flexibility of the calculator age it would seem sensible to interface a calculator module to the home computer. Many such modules can be obtained from either scrap calculators or some of the inexpensive models now available.

One chip manufacturer (National Semiconductors) is marketing a NCU chip, the MM57109, which is little more than a calculator chip disguised. None of the interface requirements have been simplified and it costs approximately £15.

The circuit and description detailed below shows the way the author tackled the problem and gives the circuit and software necessary for providing a powerful number processing unit on a home computer. The interface can be used to couple most calculator designs to any machine with one 8 bit 'OUTPUT' port and one 8 bit 'INPUT' port.

Calculators: How They Work

Before commencing on this project it is essential that the workings of a calculator, in particular the keyboard arrangements, are fully understood. To assist in the explanation Commodore Systems have kindly permitted the publishing of their 6120R calculator circuit diagram shown in Fig.1. The peculiarities of this device will be discussed later.

Many modern calculators have only one chip inside which provides both central processing and interfacing with LED's etc, they still have the same keyboard and display output conditions as the one shown.

The CPU contains an oscillator which is slow by microcomputer standards, being approx 300KHz, and all the circuitry necessary to decode inputs and carry out the required functions. This is done by using time division and sequencing techniques. A pulse of 450uS is applied to each of the 'D' leads in turn every 7.2mS as shown in Fig.2. This allows the keyboard to be arranged as a matrix, rather than having one lead per function. If, for example, the 'digit 9' key was depressed then a 450uS pulse will be presented to the pulse would be presented to the K3 input at time slot 5. At this point two special features must be taken into consideration, contact bounce and repeat entry.

In order to prevent erroneous entry due to the initial bounce of the keyboard contacts or 'fumbling' by the operator a delay is incorporated between the CPU first receiving the input condition and it acting on it, similarly when the key is released a delay is provided before another input can be recognised. These delays are approx 14mS.

The authors Nascom 1.

The display consists of a number 7 segment LED's which are sequentially illuminated under the control of the CPU 'D' leads. Each lead connects a digit display to the 8 output leads labelled A–G and DP which are coded by the CPU to light the correct segments for the value of that digit. The digits associated with each time slot are shown in the box beneath the display. IC's 3–4 are drivers which may be incorporated in the CPU.

Interface Requirements

The requirements of the interface are as follows:–

1. To buffer the TTL Programmable Input/Output port (P10) of the MPU to the CMOS devices on the calculator board.
2. To extend the instruction from the MPU O/P port to the calculator CPU during the correct time slot.
3. To present to the MPU I/P port the seven segment digit data as specified by the MPU O/P Port.

The circuit diagram of the interface is shown in Fig.3 together with its connection to an 8-bit Output Port and an 8-bit Input Port. One prime consideration is the difference in working voltages between CMOS (6V2) and the TTL (5V0). Some published circuits provide both supplies and buffer or zener down the interconnections. I have chosen to run the CMOS devices at +5V and provide buffering solely to limit drain on the CMOS input leads.

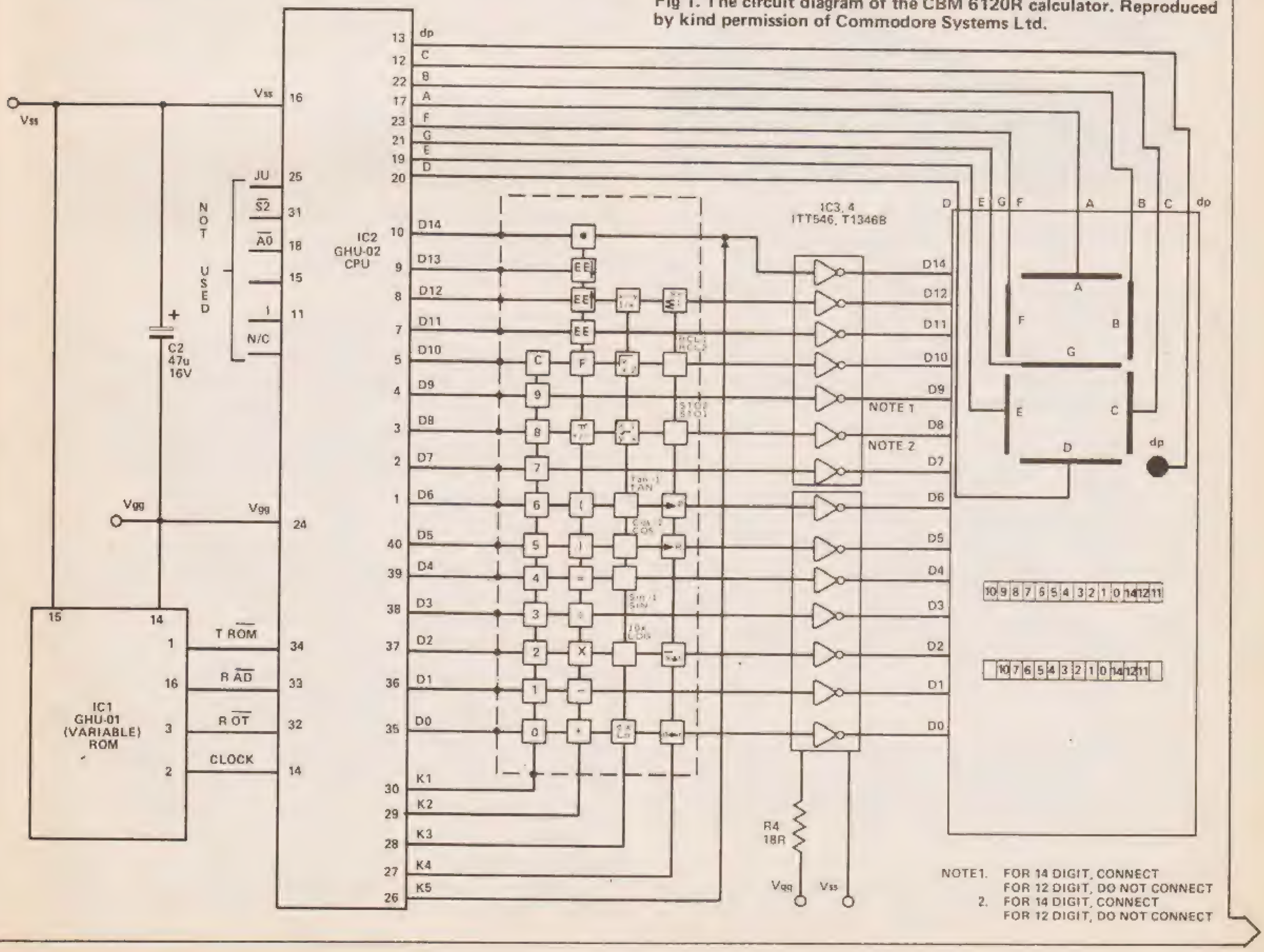
Construction Notes

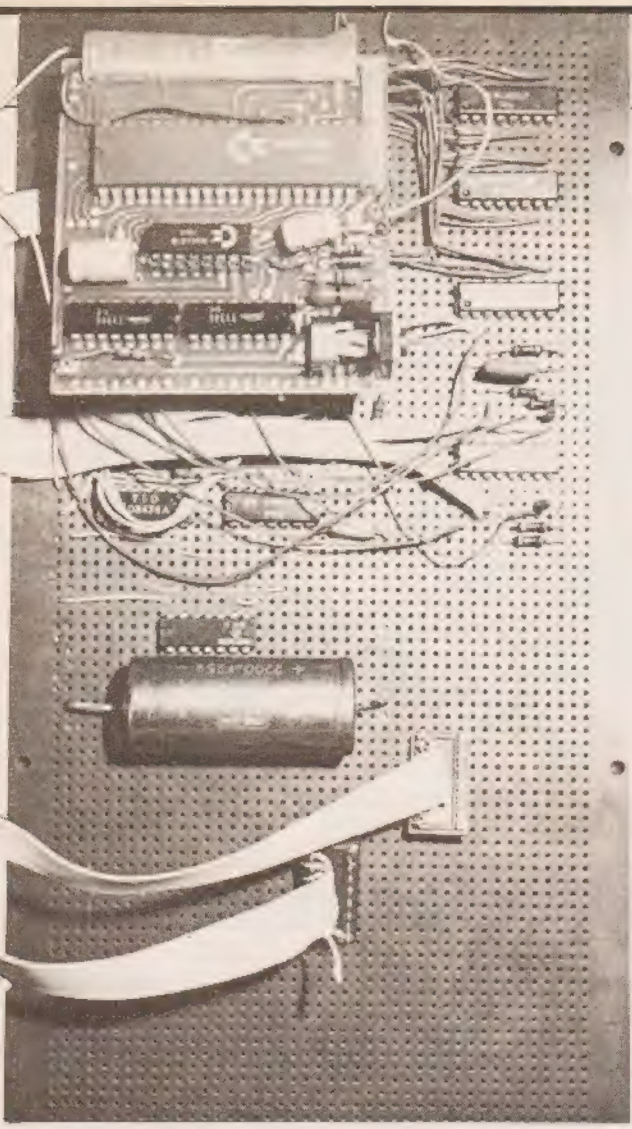
This circuit can be simply built on VERO board or a home made PCB if desired, the positioning of components is not critical, although at least three decoupling capacitors should be strategically placed across the +5V supply. The power requirements are low but it is essential that the board be fed direct from the PSU and not via the power output on the Port otherwise noise interference may upset the workings of the calculator. It is also strongly advised that the circuit of the calculator being used is obtained, or at least the key matrix format determined, before being dismantled. Once the display has been removed you are working blind.

Using A Commodore 6120R Calculator

The author chose the CBM Commodore 6120R calculator for two main reasons:– 1) It is a 48 function scientific machine which could easily be adapted for computer use, and 2) He had one! A guide to the changes necessary for other machines is given in a later paragraph.

Fig 1. The circuit diagram of the CBM 6120R calculator. Reproduced by kind permission of Commodore Systems Ltd.





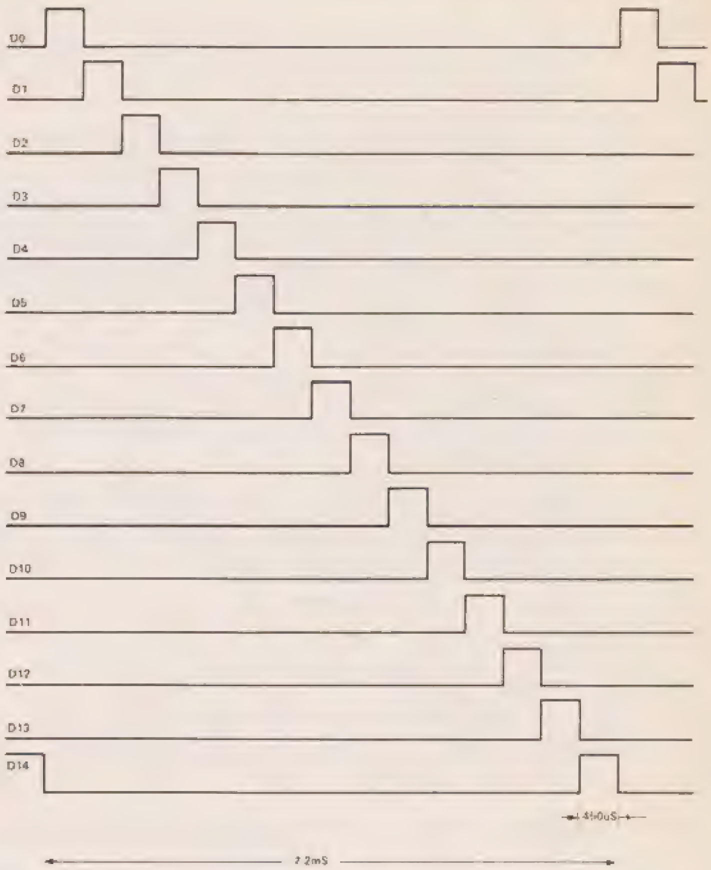


Fig 2. The data select timing diagram.

The prototype NCU on Veroboard.

The red, blue and black leads can be unsoldered from the PCB and the power and battery components removed. Now carefully remove the keyboard ribbon from the keyboard end as this can be used to connect the calculator board to the interface board. (The insulation of the ribbon melts very quickly, be warned!) The display ribbon should be removed from the PCB end and discarded. Fig.4 shows the location of the inputs and outputs from the calculator PCB (Note: the PCB itself may not be fully anotated).

Software Requirements

The listing given is for a Nascom 1 with a T2 monitor. It could, without too much difficulty, be modified for other systems. The software for the NCU is divided into two parts; Service Program which contains all the interface software for the calculator, and the Operating Program which handles the operator interface equipment e.g. to be able to access the NCU from the keyboard and get a VDU display of the problem and answer. Whilst the operating program can change from application to application, the Service Program should always be used.

Service Program

This program consists of 3 main routines which can be called from the Operating Program. Explanation of these will be brief as their operation can be deduced from the flowchart (Fig.5) and the listing.

Initialisation (INIT)

This routine, designates Port 4 as an O/P port and Port 5 as an I/P port. If preferred it can be included as part of the Operating Program. Space has been left in the listing for any special calculator stabilising sequence that may be required if start-up problems are experienced.

Input

The conditions of entry are that Register A should contain the code to be output. On return the HL register is incremented. This routine tests a code for 'second function' status and executes that subroutine if required, prior to outputting the digit to the NCU. It also waits until the calculator is ready for an input and controls the necessary timing sequences. Input codes for all calculator functions are shown in locations 0F00 onwards.

Answer (ANS)

This routine should be entered with CRT address for the answer string in HL register pair. On entry, the routine scans the first digit to determine if the answer is "ERROR" or negative. The remaining 10 digits of the argument are then scanned and printed, with all blank digits ignored. The exponent, if any, is then scanned and printed with the prefix 'E' (e.g. 6.25×10^{-12} will be displayed as 6.25E–12).

The 6120R calculator can also operate in radians. When this mode of operation is used 'rads' is displayed following the answer. In the event of an 'ERROR' the program jumps to the 'error' routine in the operating program. Location 0CAB should be changed for differing operating programs.

Operating Program

The facilities offered by this suggested program are as follows:–

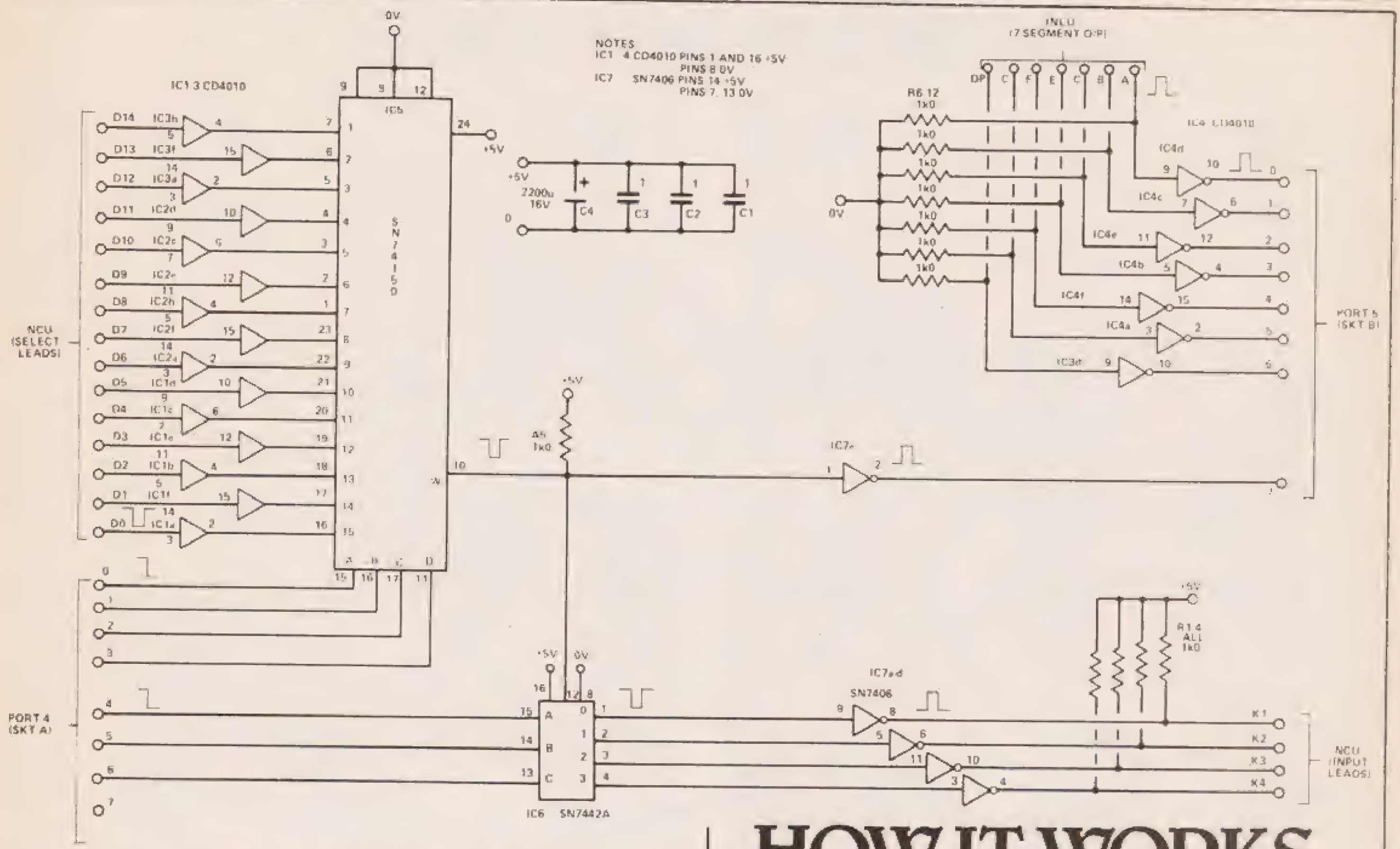


Fig.3. The circuit diagram of the NCU interface.

a) Operating instructions and error statements on the top CRT line.
b) Entry of the problem from the computer keyboard, spaces ignored.
c) Arithmetic language — i.e. as used by a calculator. 'Sin 90' = should be entered as '90 Sin =', and '−1' as '1CHS'
d) Backspace error correction.
e) Line overflow operational at any time during entry by depressing N/L key.
f) Termination of data indicated by '?' being entered.
g) Display of result commencing at the location of the '?'.

For detailed explanation of this program see Fig. 6 and the listing.

Conclusion

As stated at the beginning of this article, the use of a calculator is only an expedient until some enlightened manufacturer produces an NCU chip that is both fast, running off the computer clock, and fully computable with the address bus structure. Only then can the amount of operating software be drastically reduced.

To use other types of calculators the interface board can be left as standard or expanded to its full capacity of 16 'D' leads and 6 'K' leads if required. The software changes should be limited to the tables giving the input codes, locations marked *, which detail the display addresses and the timing loops.

HOW IT WORKS

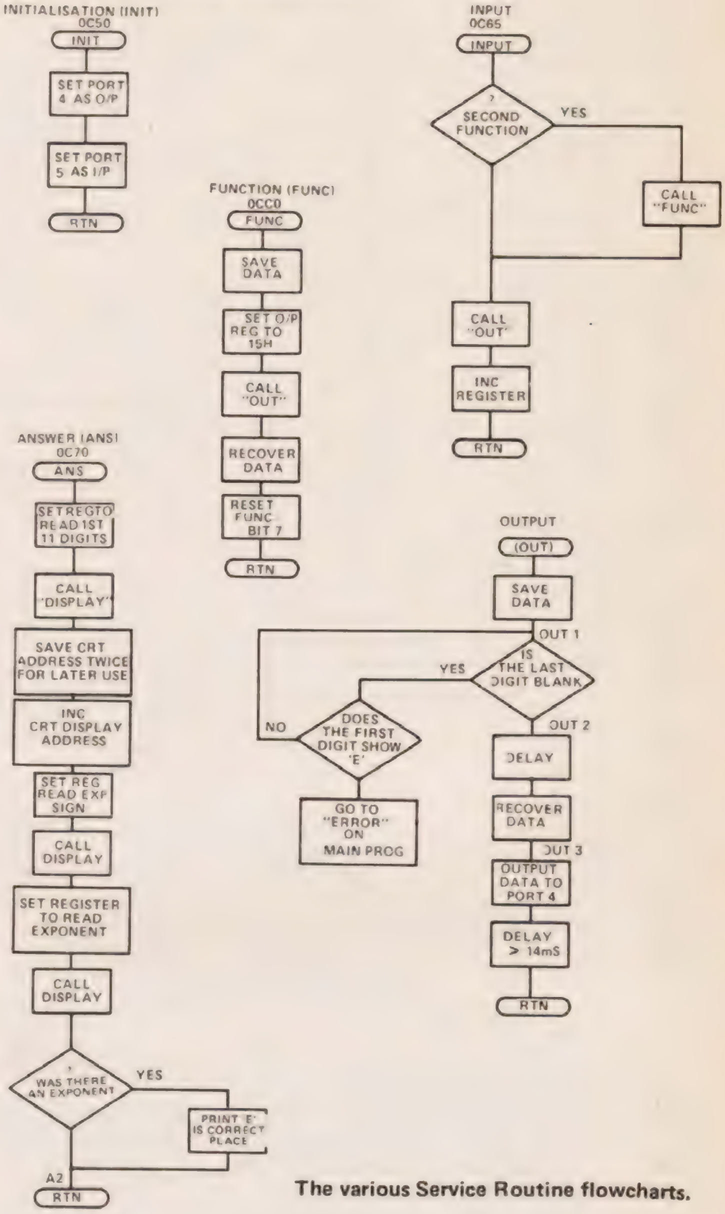
To explain how the interface is designed to work I will first describe how information is fed into the calculator. The least significant 4 bits of output port 4 are coded with the time slot information and fed direct into a data selector (IC5). This looks at the buffered 'D' Select leads from the calculator chip so that when the select lead corresponding to the binary input on the A–D leads goes high the output 'W' goes low. Bits 4, 5 and 6 of Port 4 are binary coded to indicate which of the three 'K' input leads is to be marked during the time slot for the required entry to the calculator. These two binary codings are brought together in a 4 Line-to-10 Line decoder (IC6). With the 'W' output of IC5 high, any input from Port 4 will select outputs 8 or 9 of IC6, which are both unused, but when the 'W' output goes low (i.e. during the required time slot) outputs 0–4 will go low as specified by Port 4 bits 4–6. This pulse is inverted by IC7 and pulled up by R1–4, thereby pulsing the required 'K' input lead high during the specified time slot.

To read the display the time slot associated with the required digit is binary coded onto Port 4 bits 0–3 as before. The 'W' output pulse of IC5 is inverted by IC7e to give a pulse on Port 5 pin 7 during the required time slot.

The condition present on Port 5 bits 0–6 are a buffered copy of the 7-segment outputs from the calculator CPU, note that only 6 of the 7 are required to give full decoding of digits (only 5 are required if the '6' has no top line). The program is so arranged that only the data present when bit 7 is high is considered as the valid digit. No attempt has been made at hardware decoding of the display as this is more economically carried out in software.

SERVICE ROUTINES

```
INIT
0C50   3E 4F                A=0F              Initialise P4 as O/P
0C52   D3 07                OUT 6,A
0C54   3E 0F                A=4F              Initialise P5 as I/P
0C56   D3 06                OUT 7,A
0C58   C9
'INPUT'
0C65   CB 7F                TEST B7           ?Function
0C67   C4 C0 0C             CNZ 'FUNC'
0C6A   CD 98 0C             CALL 'OUT'        Pass inst. to calculator
0C6D   23                   INC HL
0C6E   C9                   RTN
ANS*
0C70   01 45 0B             BC = 0B45         Display digit and answer length
0C73   CD D0 0C             CALL 'DISPLAY'    Read out 11 digit display
0C76   E5                   PUSH HL           Save CRT address
0C77   E5                   PUSH HL           Save CRT address
0C78   23                   INC HL            Inc. CRT
0C79*  01 41 01             BC = 0141         Exponent sign
0C7C   CD D0 0C             CALL 'DISPLAY'
0C7F*  01 43 02             BC = 0243         Exponent value
0C82   CD D0 0C             CALL 'DISPLAY'
0C85   37                   SCF
0C86   CI                   POP BC            Was there an exponent—
0C87   ED 42                SBC BC
0C89   28 05                JRZ 'A1'          If no go to 'A1'
0C8B   E1                   POP HL            Recover CRT address
0C8C   36 45                (HL), 45          Print 'E'
0C8E   18 01                JR 'A2'
0C90   C1          'A1'     POP BC            Waste CRT address
0C91   C9          'A2'     RTN
Subroutines
'OUT'
0C98   F5                   PUSH AF           Save Data
0C99*  3E 4F       'OUT 1'  A = 4F
0C9B   D3 04                OUT 4,A           Test LSB to await calculator
0C9D   DB 05                IN A,5            to be ready
0C9F   FE 81                CP = 81
0CA1   30 0D                JNC 'OUT 2'       If not blank go to 'out 2'
0CA3*  3E 45                A = 45            Otherwise test for error
0CA5   D3 04                OUT 4,A
0CA7   DB 05                IN A,5
0CA9   FE B9                CP = B9           Test Error
0CAB   CA E0 0E             JZ – 'ERROR'
0CAE   18 E9                JR 'OUT'          If still blank do again
0CB0   FF          'OUT 2'  RST 56            Delay
0CB1   F1                   POP AF            Recover Data
0CB2   D3 04       'OUT 3'  OUT 4,A           Pass data to calculator
0CB4   FF                   RST '56'
0CB5   FF                   RST '56'          Delay > 14m Sec for contact
0CB6   FF                   RST '56'          bounce etc.
0CB7   C9                   RTN
'FUNC'
0CC0   F5                   PUSH AF           Save data
0CC1   3E 15                A = 15            Code for 'func'
0CC3   CD 98 0C             CALL 'OUT'        Output function data
0CC6   F1                   POP AF            Recover data
0CC7   CB BF                R/S B7            Reset function bit
0CC9   C9                   RTN
```



The various Service Routine flowcharts.

DISPLAY

```
0CD0   79          'D1'     A,C               Load digit number/time slot
0CD1   D3 04                OUT 4, A          Output time slot
0CD3   DB 05       'D2'     IN A, 5           Input display
0CD5   CB 7F                TEST A7           ? Correct time slot
0CD7   28 FA                JRZ 'D2'          If no do again
0CD9   DB 05                IN A, 5           Double check input
0CDB   FE 80                CP = 80
0CDD   38 F1                JRC 'D'           If cleared do again
0CDF   20 04                JRNZ 'D4'         If not blank jump to 'D4'
0CE1   0C          'D3'     INC C             Step to next digit and do again
0CE2   10 EC                DJNZ 'D1'         if not finished
0CE4   C9                   RTN               Return if finished
0CE5   FE C0       'D4'     CP = C0           Set carry if no dec. pt
0CE7   28 22                JRE – D6          If blank – d.p. jump to 'D6'
0CE9   CB B7                R/S A6            Reset dp bit (bit 6)
0CEB   D9                   EXX
0CEC   01 15 00             BC = 0015
0CEF   21 20 0D             HL = 0D20         Search table for match
0CF2   ED B1                CP1R
0CF4   C2 E0 0E             JNZ 'ERROR'       If match not found
0CF7   7E                   A, (HL)           Load translation
0CF8   D9                   EXX
0CF9   77                   (HL), A           Print digit
0CFA   23                   INC HL            Step to next CRT locn
0CFB   38 E4                JRC 'D3'          If no dp go for next digit
0CFD   3E 01                A = 01
0CFF   B8                   CP B              If last digit supress decimal
0D00   28 05                JRZ 'D5'          point
0D02   36 2E                (HL), 2E          Print dp.
0D04   23                   INC HL            Step to next CRT locn.
0D05   18 DA                JR 'D3'           Go for next digit
0D07   79          'D5'     A, C              Does point follow the LSD
0D08   FE C4                CP = C1           of exponent?
0D0A   C0                   RNZ               RTN if no.
0D0B   E5          'D6'     PUSH HL           If yes save CRT address
0D0C   22 18 0C             (DC18), HL
0D0F   EF 20 20 72 61 64 73 00                Print 'Rads'
0D17   E1                   POP HL            Recover CRT address
0D18   C9                   RTN               Return
DIGIT DISPLAY TABLE
0D20   9F 30 86 31
0D24   AB 32 A7 33
0D28   B6 34 B5 35
0D2C   BD 36 87 37
0D30   BF 38 B7 39
0D34   A0 2D
```

OPERATING PROGRAM

Address	Code	Label	Instruction	Comment
0E00	CD 50 0C		CALL 'S/INIT'	Initialise
0E03	EF 1E 00			Clear CRT
0E06	DD 21 D0 0B		1X = 0BD0	
0E0A	DD 22 18 0C		(0C18), (1X)	Set cursor
0E0E	EF		PRINT	
	ENTER DATA NOW		TEXT	
0E1D	00			
0E1E	21 10 09		HL = 0910	Set CRT address
0E21	22 00 08	'F1'	(0800), (HL)	Save line address in temp reg.
0E24	22 18 0C		(0C18), (HL)	Set cursor
0E27	CD 3E 00	'F2'	CALL 'CHIN'	Get entry
0E2A	FE 1F		CP = 1F	? N/L
0E2C	20 10		JRNZ 'F3'	If not a new line F3
0E2E	3E 09		A = 09	
0E30	CD 3B 01		CALL 'CRT'	Print ' '
0E33	2A 00 08		(HL), (0800)	Recover line address
0E36	01 7F 00		BC = 007F	Set HL to next but one
0E39	09		ADD HL, BC	line – 1 place
0E3A	77		(HL), A	Print ' '
0E3B	23		INC HL	Step CRT
0E3C	18 E3		JR 'F1'	Go for next digit
0E3E	CD 3B 01	'F3'	CALL 'CRT'	Print the digit
0E41	FE 3F		CP = 3F	Cp = '?' End of data
0E43	20 E2		JRNZ 'F2'	If no go for next digit
0E45	2A 18 0C		HL, (0C18)	Load cursor address into HL
0E48	2B		DEC HL,	Decrement to cover '?'
0E49	DD 22 18 0C		(0C18), (1X)	Set CRT address
0E4D	EF		PRINT	
	CALCULATING NOW		TEXT	
0E5D	00			
0E5E	21 10 09		HL = 0910	Reset line address store to
0E61	22 00 08		(0800), (HL)	the first line
0E64	18 01		JR 01	Miss the next locn.
0E66	23	'F4'	INC HL	Step to next CRT locn.
0E67	7E	'F5'	A, (HL)	Load data into A
0E68	FE 20		CP = 20	
0E6A	28 FA		JRZ 'F4'	Ignore spaces
0E6C	FE 09		CP = 09	? N/L
0E6E	20 0C		JRNZ – 'F6'	If no go to F7
0E70	2A 00 08		HL, (0800)	Record line address
0E73	01 80 00		BC = 0075	Step to next line and
0E76	09		ADD HL, BC	store
0E77	22 00 08		0800, (HL)	
0E7A	18 EB		JR 'F5'	
0E7C	FE 3F	'F6'	CP = 3F	? All data entered
0E7E	20 06		JRNZ – 'F7'	If no jump
0E80	CD 70 0C		CALL 'S/ANS'	Call service program
0E83	C3 B1 0E		JP 'END'	
0E86	FE 45	'F7'	CP – 45	CP = 'E'
0E88	28 0F		JRZ – 'F9'	
0E8A	FE 41		CP41	? Numerical
0E8C	38 0B		JRC – 'F9'	If yes go to F9
0E8E	23		INC HL	Step to next CRT point
0E8F	86		ADD (HL)	Add 2nd letter to first
0E90	F5		PUSH AF	Save resultant data
0E91	3E 41		A = 41	
0E93	23	'F8'	INC HL	Search for end of
0E94	BE		CP (HL)	word string
0E95	38 FC		JRC 'F8'	
0E97	2B		DEC HL	
0E98	F1		POP AF	Recover data
0E99	E5	'F9'	PUSH HL	Save CRT address
0E9A	21 FE 0E		HL = EFE	Table address – 2
0E9D	0E 31		C = 31	Table length
0E9F	23	'F10'	INC HL	Step to next pair of
0EA0	23		INC HL	data
0EA1	BE		CP (HL)	Is there a match?
0EA2	28 05	'F11'	JRZ 'F12'	If yes go to F12
0EA4	0D		DEC C	Decrement couple
0EA5	20 F8		JRNZ 'F10'	Repeat till match found
0EA7	18 37		JR 'ERROR'	If not found error
0EA9	23		INC HL	Load translation
0EAA	7E		A, (HL)	
0EAB	CD 65 0C		CALL 'S/INPUT'	Enter into NCU
0EAE	E1		POP HL	Recover CRT address
0EAF	18 B5		JR F4	Do for next entry
END				
0EB1	DD 22 18 0C		(0C 18 1X)	Set cursor
0EB5	EF	'E1'	'PRINT'	
0EB6	PUSH BAR TO RE–ENTER		TEXT	
0ECA	00			
0ECB	CD 3E 00	'E2'	CALL 'CHIN'	
0ECE	FE 20		CP = 20	Wait for entry
0ED0	20 F9		JRNZ 'E2'	
0ED2	31 00 10		SP = 0100	Reset stack point
0ED5	21 03 0E		HL = 0E00	Enter return address
0ED8	E5		PUSH HL	onto top of stack
0ED9	3E 05		A = 05	Load clear data
0EDB	C3 B2 0C		JP 'S/OUT 3'	Jp to execute the clear
ERROR				
0EE0	DD 22 18 0C		(0C18), (1X)	Set cursor
0EE4	EF		'PRINT'	
0EE5	ERROR!!//			Print ERROR!! + 2 spaces
0EEE	00			
0EEF	18 C4		JR 'E1'	

TABLE

F00 TO F5F

Code		Key	Function
30	0F	0	
31	0E	1	
32	0D	2	
33	0C	3	
34	0B	4	
35	0A	5	
36	09	6	
37	08	7	
37	07	8	
39	06	9	
2B	1F	+	
2D	1E	—	
2A	1D	*	Multiply
2F	1C	/	Divide
3D	1B	=	
29	1A	)	
28	19	(	
2E	11	.	Decimal point
45	14	E	Exponent
40	1D	@	
27	27	'	x^y
22	25	"	x^2
2C	23	,	1/x
A4	A5	SQR	$\sqrt{x}$

Code		Key	Function
A1	A7	ROOT	$x\sqrt{y}$
99	97	Pi	π
B1	A3	XY	X <> Y
96	3F	DR	Deg <> Rads
8B	17	CHS	Change sign
9C	2B	SIN	
94	AB	ASIN	Sin^{-1}
92	2A	COS	
84	AA	ACOS	Cos^{-1}
A2	29	TN	Tangent
95	A9	ATN	Tan^{-1}
93	2D	Lg.	Log^{10}
8D	AD	AL	Antilog
9A	2F	LN	Logn
8F	AF	AN	e^X
A6	B3	XN	Statistic data entry
97	3D	AV	Average
7D	3A	P–R	Polar to rectangular
7F	39	R–P	Rectangular to Polar
7A	37	I1	In memory 1
7B	B7	I2	In memory 2
80	35	O1	Out memory 1
81	B5	O2	Out memory 2
8E	33	E1	Sum in memory 1
A5	05	RST	Reset

computing today

WHAT TO LOOK FOR IN THE NOVEMBER ISSUE. ON SALE OCTOBER 12TH

PET IMPRESSIONS

Like the TRS-80 article in this issue we present another "first time" buyers look at the Commodore PET. Written by Mr Stephenson (of D2 programming fame) it investigates the pro's and con's of buying a home micro and the impressions it has created. An article not to be missed if you are considering buying a PET.

GRAPHICALLY PROJECTED

A graphics project for your Nascom is this months offering. Requiring the minimum disruption to the processor card it is a simple project to build and uses pixcels rather than pre-determined characters. These give greater flexibility to the system and allow you to easily program it. Grab your soldering iron and stand by for this one.

LIVING WITH A GAME

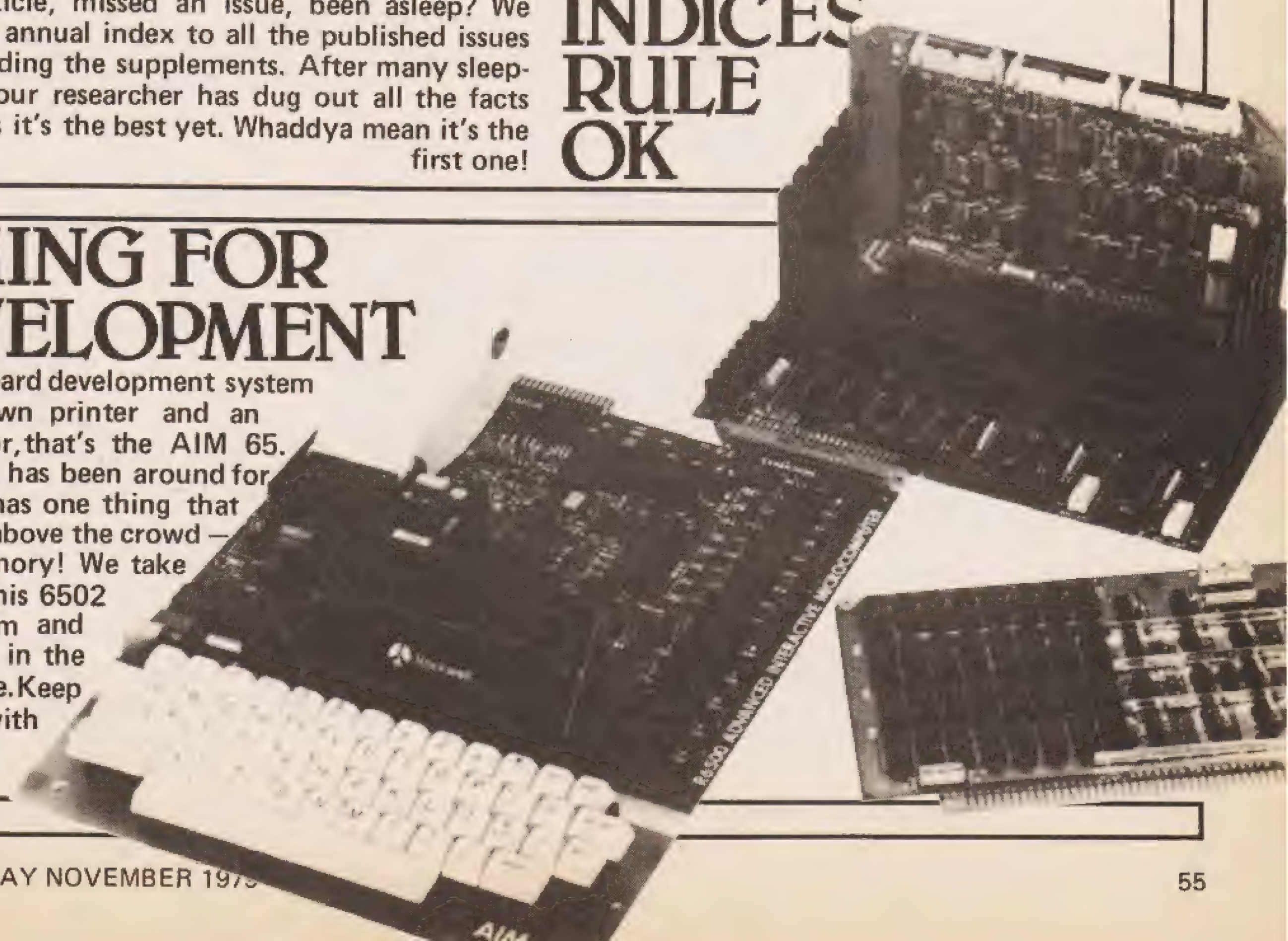
The famous game of Life, developed by Conway – oresented by Kuczinski, will have you racking your brains for the ultimate pattern in the struggle to survive. A complete object code listing for the Nascom 1 is supplied along with a sample run to show game in action. Will you survive the wait till next months issue?

INDICES RULE OK

Lost an article, missed an issue, been asleep? We present our annual index to all the published issues of CT including the supplements. After many sleepless nights our researcher has dug out all the facts and reckons it's the best yet. Whaddya mean it's the first one!

AIMING FOR DEVELOPMENT

A single board development system with it's own printer and an 8K monitor, that's the AIM 65. Although it has been around for a while it has one thing that elevates it above the crowd – bubble memory! We take a look at this 6502 based system and its position in the marketplace. Keep on target with our system review.



Trevor Lusty

We improve last months trial solution in a variety of ways, and give you a new one to ponder

Last month, the problem was to find a house number in a row of odd numbered houses, such athat the sum of the house numbers on one side equalled the sum on the other side. A possible solution was given but, although it gives a correct answer, the program (listing 1) is a rather poor attempt.

```
100 REM  *****  MY UNCLE LIVES IN HOUSE - N  *****
120 LET N=1
130 LET N=N+2
140 REM  *****  SUM HOUSE NUMBERS ON LEFT IN - L  *****
150 LET L=0
160 FOR I=1 TO N-2 STEP 2
170 LET L=L+I
180 NEXT I
190 REM  *****  SUM HOUSE NUMBERS ON RIGHT IN - R  *****
200 REM  *****  STARTING AT HOUSE NUMBER - H       *****
210 LET H=N
220 LET R=0
230 LET H=H+2
240 LET R=R+H
250 REM  *****  DOES THE LEFT SUM EQUAL THE RIGHT SUM ?  *****
260 IF L=R THEN 300
270 IF L>R THEN 230
280 GOTO 130
290 REM  *****  HOW MANY HOUSES ?  *****
300 LET T=(H+1)/2
310 IF T>100 THEN 330
320 GOTO 130
330 PRINT "THERE ARE";T;"HOUSES IN THE ROAD"
340 PRINT "UNCLE LIVES AT HOUSE NUMBER";N
350 END


THERE ARE 169 HOUSES IN THE ROAD
UNCLE LIVES AT HOUSE NUMBER 239
```

Fig.1. Last months solution.

What Is Better?

Before we can 'improve' a program, we must ask ourselves just what we mean by 'better'. An ideal program should find the correct solution and execute quickly without ridiculous storage requirements. The given program is correct and uses little storage, so better here means faster! The program takes 76 seconds to execute on a PET. This may not seem too bad, but for longer roads the time required increases dramatically.

A small, but significant, improvement can be achieved by tidying up the given program. The first thing to note is that, in lines 260 and 270, the left sum — 'L' is more likely to be greater than the right sum — 'R' than equal to it. Thus reversing these two lines means that the L = R test is not carried out unnecessarily. Another small improvement is to eliminate the GOTO statements.

eg. 310 IF T > 100 THEN 330
320 GOTO 130

may be replaced by the single statement :—

310 IF T < = 100 THEN 130

With these improvements, the program (listing 2) takes 68 seconds to execute ——— this is 10.5% faster than the original version.

```
100 REM  *****  MY UNCLE LIVES IN HOUSE - N  *****
120 LET N=1
130 LET N=N+2
140 REM  *****  SUM HOUSE NUMBERS ON LEFT IN - L  *****
150 LET L=0
160 FOR I=1 TO N-2 STEP 2
170 LET L=L+I
180 NEXT I
190 REM  *****  SUM HOUSE NUMBERS ON RIGHT IN - R  *****
200 REM  *****  STARTING AT HOUSE NUMBER - H       *****
210 LET H=N
220 LET R=0
230 LET H=H+2
240 LET R=R+H
250 REM  *****  DOES THE LEFT SUM EQUAL THE RIGHT SUM ?  *****
260 IF L>R THEN 230
270 IF L<>R THEN 130
290 REM  *****  HOW MANY HOUSES ?  *****
300 LET T=(H+1)/2
310 IF T<=100 THEN 130
330 PRINT "THERE ARE";T;"HOUSES IN THE ROAD"
340 PRINT "UNCLE LIVES AT HOUSE NUMBER";N
350 END


THERE ARE 169 HOUSES IN THE ROAD
UNCLE LIVES AT HOUSE NUMBER 239
```

Fig.2. A slightly improved version.

Improving The Method

For further improvements we must look at the algorithm. The method used requires that we calculate the left sum and the right sum from scratch for each new value of N. This is totally unnecessary ——— when we move the test house, the new left sum is merely increased by adding on the old test value. Similarly, the right sum must be decreased by the new test value.

eg. Imagine that the test for house 11 has just failed.
The next test is set up as follows:—

Left Sum	Test Value		Right Sum
25 = 1+3+5+7+9	11	13+15	= 28
36 = 1+3+5+7+9+11	13	15	= 15
		15+17	= 32
		15+17+19	= 51

```
100 REM  *****  MY UNCLE LIVES IN HOUSE - N  *****
120 LET N=1
130 LET L=0
140 LET R=3
145 LET H=3
150 REM  *****  MOVE HOUSE AND CHANGE LEFT AND RIGHT SUMS  *****
160 LET L=L+N
170 LET N=N+2
180 LET R=R-N
230 LET H=H+2
240 LET R=R+H
245 IF L>R THEN 230
250 REM  *****  DOES THE LEFT SUM EQUAL THE RIGHT SUM ?  *****
270 IF L<>R THEN 160
290 REM  *****  HOW MANY HOUSES ?  *****
300 LET T=(H+1)/2
310 IF T<=100 THEN 160
330 PRINT "THERE ARE";T;"HOUSES IN THE ROAD"
340 PRINT "UNCLE LIVES AT HOUSE NUMBER";N
350 END


THERE ARE 169 HOUSES IN THE ROAD
UNCLE LIVES AT HOUSE NUMBER 239
```

Fig.3. A better solution but still not the best.

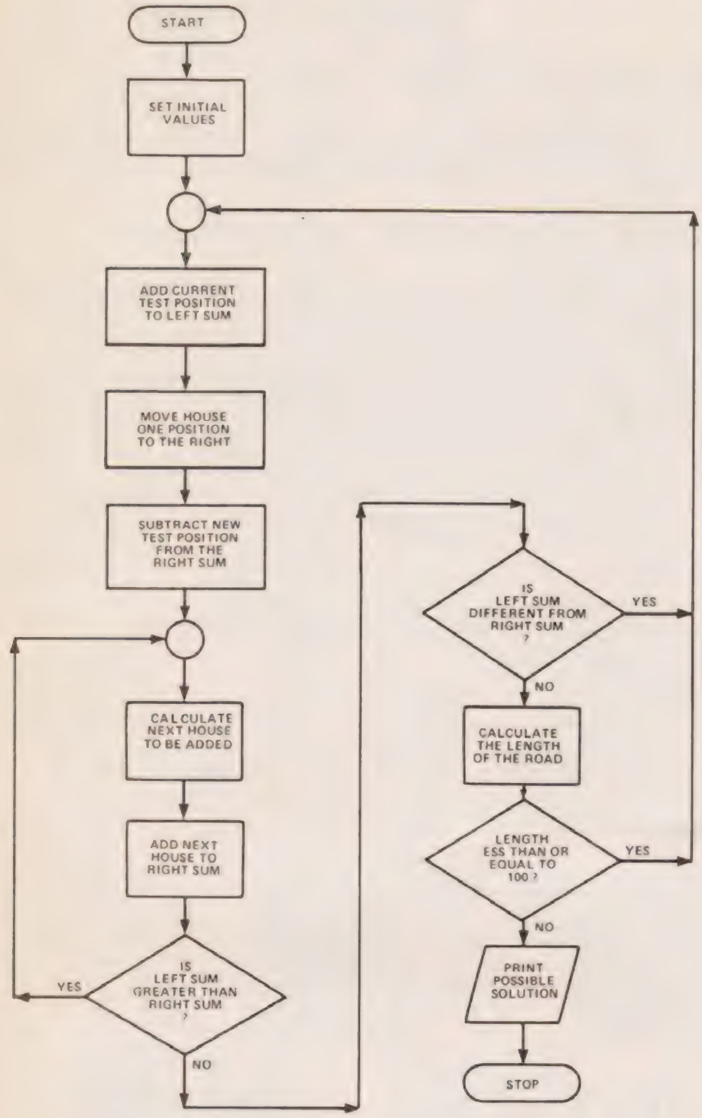


Fig.4. The flowchart for the best solution to the problem.

```
10 REM ********************************
20 REM *                              *
30 REM * PROGRAM ---   'WHAT-HOUSE'   *
35 REM *     BY RECURRENCE FORMULA    *
40 REM *                              *
50 REM * PROGRAMMED IN 'PET' BASIC    *
60 REM *                              *
70 REM * TREVOR L LUSTY    12/8/79.   *
80 REM *                    .         *
90 REM ********************************
95 REM
2000 REM  *****  SET INITIAL VALUES  *****
2020 LET M1 = 1
2040 LET M2 = 7
2060 REM  *****  COMPUTE NEXT VALUE  *****
2080 LET N = 6*M2 - M1
2100 REM  *****  MOVE VALUES   *****
2120 LET M1 = M2
2140 LET M2 = N
2160 REM  *****  COMPUTE LENGTH OF ROAD  *****
2180 LET L = INT(SQR( (N*N+1)/2 )+.5)
2200 REM  *****  PRINT POSSIBLE SOLUTION  *****
2220 PRINT
2240 PRINT
2260 PRINT "THERE ARE";L;"HOUSES"
2280 PRINT "UNCLE LIVES IN NUMBER";N
2300 REM  *****  IS IT TIME TO STOP ?  *****
2320 IF N<10000000 THEN 2080
2340 END
```

```
THERE ARE 29 HOUSES
UNCLE LIVES IN NUMBER 41

THERE ARE 169 HOUSES
UNCLE LIVES IN NUMBER 239

THERE ARE 985 HOUSES
UNCLE LIVES IN NUMBER 1393

THERE ARE 5741 HOUSES
UNCLE LIVES IN NUMBER 8119

THERE ARE 33461 HOUSES
UNCLE LIVES IN NUMBER 47321

THERE ARE 195025 HOUSES
UNCLE LIVES IN NUMBER 275807

THERE ARE 1136689 HOUSES
UNCLE LIVES IN NUMBER 1607521

THERE ARE 6625109 HOUSES
UNCLE LIVES IN NUMBER 9369319

THERE ARE 38613965 HOUSES
UNCLE LIVES IN NUMBER 54608393
```

Fig.5. The best solution to the problem.

The program for this method is given in listing 3 and a dry run through the associated flowchart should clarify the procedure.

When this method is programmed it takes only 5 seconds to execute ––– an improvement of 93.4%. It is sufficiently fast to allow more values to be computed within a reasonable time, but it's still not the best solution.

Leading From Behind!

When we look at a number of solutions we find that they are connected by a recurrence formula. (A recurrence formula is a way of finding the next value in a sequence from the preceding values.) In this case, the formula for the middle house is :–

(NEXT VALUE) = 6 times (THIS VALUE) – (LAST VALUE)

or, in more normal mathematical notation :–

$$R_{n+1} = 6R_n - R_{n-1}$$

Given that the first solution is 1 ––– zero sum on either side, and the second solution is 7 ––– 1+3+5 = 9. The next may be calculated from 6x7 – 1 = 41.

The solution we found is calculated next 6x41 – 7 = 239. The program for this method is given in listing 4 and, as the values are printed immediately, it might be said to represent almost 100% improvement on the original program. As this is a computing magazine and not a mathematical one, an algebraic proof of the recurrence formula would be out of place. However, if any reader can find a geometric proof I would be pleased to see it.

Problem Of The Month

Write a BASIC program (it can even be done in integer BASIC) to divide one number by another and printout the answer correct to any specified (unlimited) number of decimal places.

electronics today international

What to look for in the December Issue: On sale November 2nd

10 PROJECT SPECIAL

Ten (10), yes ten projects for you — everything from a flash trigger, for all the budding Patrick Lichfields out there, to a versatile 1 minute to 20 hours timer. There's something for everyone — a rain alarm to warn you when it's welly time and a hum filter to give your hi-fi for Christmas.

AIRCRAFT BAND CONVERTER

Use one of the terrific ten, the ETI Biggles detector, with your short wave set to listen in to the world's airlines. As a Jumbo (jet, not hephalump) screams overhead on it's way to New York, Kuwait or Staine's High Street, you can hear the pilot chatting to air traffic control in his strange aero jargon.

FUNCTION GENERATOR

Or. . . .
Short of a few functions for your project bench? Roll up, roll up — get your functions here in all shapes and sizes — sine, square and triangle. The generator comes complete with analogue meter — no calibration problems here.

CHESS COMPUTERS

Forget the lonely hearts column. Find your perfect partner for the long winter evenings among the latest chess computers. Chess Mate will give you it's undivided attention for as long as it takes you to mate and Boris will beat you off the board (it takes all sorts). We'll also have news of the long-awaited speaking chess computer.

computing today

technical book service

What Is A Microprocessor ?
2 Cassette tapes plus a 72 page book deal with many aspects of microprocessors including Binary and Hexadecimal counting, Pro— gramming etc. £10.75.

Adams, C. BEGINNERS GUIDE TO COMPUTERS AND MICROPROCESSORS WITH PROJECTS £5.60
Understanding building programming and operating your own microcomputer.

Ahl, BASIC COMPUTER GAMES £5.25

Albrecht, B. BASIC FOR HOME COMPUTERS. A self teaching guide £4.75
Shows you how to read, write and understand basic programming language used in the new personal size microcomputers.

Albrecht B. BASIC. A self teaching guide (2nd edition) £4.50
Teach yourself the programming language BASIC. You will learn how to use the computer as a tool in home or office and you will need no special maths or science background.

Alcock, D. ILLUSTRATING BASIC £2.60
This book presents a popular and widely available language called BASIC, and explains how to write simple programs.

Altman, I. MICROPROCESSORS £10.65
Gives a general overview of the technology design ideas and explains practical applications.

Altman, L. APPLYING MICROPROCESSORS £12.00
Follow volume which takes you into the second and third generation devices.

Aspinall, D. INTRO TO MICROPROCESSORS £6.40
Explains the characteristics of the component.

Barden, W. Z-80 MICROCOMPUTER HANDBOOK £7.65

Barden, W. HOW TO BUY AND USE MINICOMPUTERS AND MICROCOMPUTERS £7.75
Discusses these smaller computers and shows how they can be used in a variety of practical and recreational tasks in the home or business.

Barden, W. HOW TO PROGRAM MICROCOMPUTERS £7.00
This book explains assembly language programming of microcomputers based on the Intel 8080, Motorola MC6800 and MOS Technology MCS6502 microprocessor.

Barna, A. INTRODUCTION TO MICROCOMPUTERS AND MICROPROCESSORS £8.15
Provides the basic knowledge required to understand microprocessor systems. Presents a fundamental discussion of many topics in both hardware and software.

Bibbero, R. J. MICROPROCESSORS IN INSTRUMENTS AND CONTROL £12.45
Introduces the background elements, paying particular regard to the dynamics and computational instrumentation required to accomplish real-time data processing tasks.

Lancaster, D. TV TYPEWRITER COOKBOOK £7.75
An in-depth coverage of tv typewriters (tv's) the only truly low cost microcomputer and small display interface.

Lancaster, D. CHEAP VIDEO COOKBOOK £6.50

Lesea, A. MICROPROCESSOR INTERFACING TECHNIQUES £8.50

Leventhal. INTRO TO MICROPROCESSORS £16.70

Lewis, T. G. MIND APPLIANCE HOME COMPUTER APPLICATIONS £4.75

Libes, S. SMALL COMPUTER SYSTEMS HANDBOOK £5.75
The Primer written for those new to the field of personal home computers.

Lippiatt. ARCHITECTURE OF SMALL COMPUTER SYSTEMS £4.35

Moody, R. FIRST BOOK OF MICROCOMPUTERS £3.85
(the home computer owners best friend).

McGlynn, D. R. MICROPROCESSORS — Technology, Architecture & Applications £9.00
This introduction to the 'computer-on-a-chip' provides a clear explanation of the important new device.

McMurran, PROGRAMMING MICROPROCESSORS £5.50
A practical programming guide that includes architecture, arithmetic/logic operations, fixed and floating point computations, data exchange with peripheral devices computers and other programming aids.

Monro, INTERACTIVE COMPUTING WITH BASIC £3.65

Nagin, P. BASIC WITH STYLE £4.00
Programming Proverbs. Principles of good programming with numerous examples to improve programming style and producing.

Ogdin SOFTWARE DESIGN FOR MICROCOMPUTERS £7.00

Ogdin. MICROCOMPUTER DESIGN £7.05

Peatman, MICROCOMPUTER BASE DESIGN £5.45

Peatman, J. B. MICROCOMPUTER BASED DESIGN £20.40
This book is intended for undergraduate courses on microprocessors.

Peckham, HANDS ON BASIC WITH A PET **£8.70**

Peckham, BASIC — A HANDS ON METHOD **£6.85**

Bursky, D. MICROCOMPUTER BOARD DATA MANUAL **£5.40**

Bursky, D. MICROPROCESSOR DATA **£5.40**

Includes complete description of the processor. Support circuits, Architecture, Software, etc.

Coan, J. S. BASIC BASIC **£7.50**

An introduction to computer programming in BASIC language.

Coan, J. S. ADVANCED BASIC **£7.30**

Applications and problems.

Ditlea, A SIMPLE GUIDE TO HOME COMPUTERS **£4.00**

Freiberger, S. CONSUMERS GUIDE TO PERSONAL COMPUTING AND MICROCOMPUTERS **£5.50**

Frenzel, L. GETTING ACQUAINTED WITH MICROPROCESSORS **£7.10**

This is an invaluable book for those who want to know more about hobby and personal computing.

Gilmore, C. M. BEGINNERS GUIDE TO MICROPROCESSORS **£4.75**

Grossworth, BEGINNERS GUIDE TO HOME COMPUTERS **£3.10**

Gosling, R. E. BEGINNING BASIC **£3.25**

Introduces BASIC to first time users.

Graham, N. MICROPROCESSOR PROGRAMMING FOR COMPUTER HOBBYISTS **£7.00**

Haviland, N. P. THE COMPULATOR BOOK **£6.20**

Building super calculators and minicomputer hardware with calculator chips.

Hartley, INTRODUCTION TO BASIC **£2.40**

Heiserman, D. L. MINIPROCESSORS FROM CALCULATORS TO COMPUTERS **£4.85**

Hilburn, J. L. MICROCOMPUTERS, MICROPROCESSORS, HARDWARE, SOFTWARE AND APPLICATIONS **£16.95**

Complete and practical introduction to the design, programming operation, uses and maintenance of modern microprocessors, their integrated circuits and other components.

Klingman, E. MICROPROCESSOR SYSTEMS DESIGN **£16.95**

Outstanding for its information on real microprocessors, this text is both an introduction and a detailed information source treating over a dozen processors, including new third generation devices. No prior knowledge of microprocessors or microelectronics is required for the reader.

Kemeny, J. G. BASIC PROGRAMMING **£6.55**

A basic text.

Korn, G. A. MICROPROCESSOR AND SMALL DIGITAL COMPUTER SYSTEMS FOR ENGINEERS AND SCIENTISTS **£19.00**

This book covers the types, languages, design software and applications of microprocessors.

Duncan. MICROPROCESSOR SOFTWARE ENGINEERING **£13.50**

Rao, G. U. MICROPROCESSOR AND MICROPROCESSOR SYSTEMS **£20.50**

A completely up-to-date report on the state-of-the-art of microprocessors and microcomputers written by one of the leading experts.

Rony, P. H. THE 8080A BUGBOOK: Microcomputer Interfacing & Programming **£8.15**

The principles, concepts and applications of an 8-bit microcomputer based on the 8080 microprocessor IU chip. The emphasis is on a computer as a controller.

Scelbi. 6800 SOFTWARE GOURMET GUIDE AND COOKBOOK **£8.80**

Scelbi. 8080 SOFTWARE GOURMET GUIDE AND COOKBOOK **£8,80**

Scelbi. UNDERSTANDING MICROCOMPUTERS **£8.60**

Gives the fundamental concepts of virtually all microcomputers.

Spencer, GAME PLAYING WITH BASIC **£4.70**

Schoman, K. THE BASIC WORKBOOK **£3.70**

Creative techniques for beginning programmers.

Sirion, D. BASIC FROM THE GROUND UP **£6.00**

Soucek, B. MICROPROCESSORS AND MICROCOMPUTERS **£19.00**

Here is a description of the applications programming and interfacing techniques common to all microprocessors.

Spracklen, D. SARGON **£9.75**

A computer chess program in Z-80 assembly language.

Titus, MICROCOMPUTER ANALOGUE CONVERTER **£7.45**

Titus, 8080/8085 SOFTWARE DESIGN **£7.45**

Tracton. 57 PRACTICAL PROGRAMS & GAMES IN BASIC **£6.40**

Programs for everything from Space war games to Blackjack.

Waite. M. MICROCOMPUTER PRIMER **£6.25**

Waite, YOUR OWN COMPUTER **£1.50**

Introduces the beginner to the basic principles of the microcomputer.

Ward. MICROPROCESSOR / MICROPROGRAMMING HANDBOOK **£6.00**

Authoritative practical guide to microprocessor construction programming and applications.

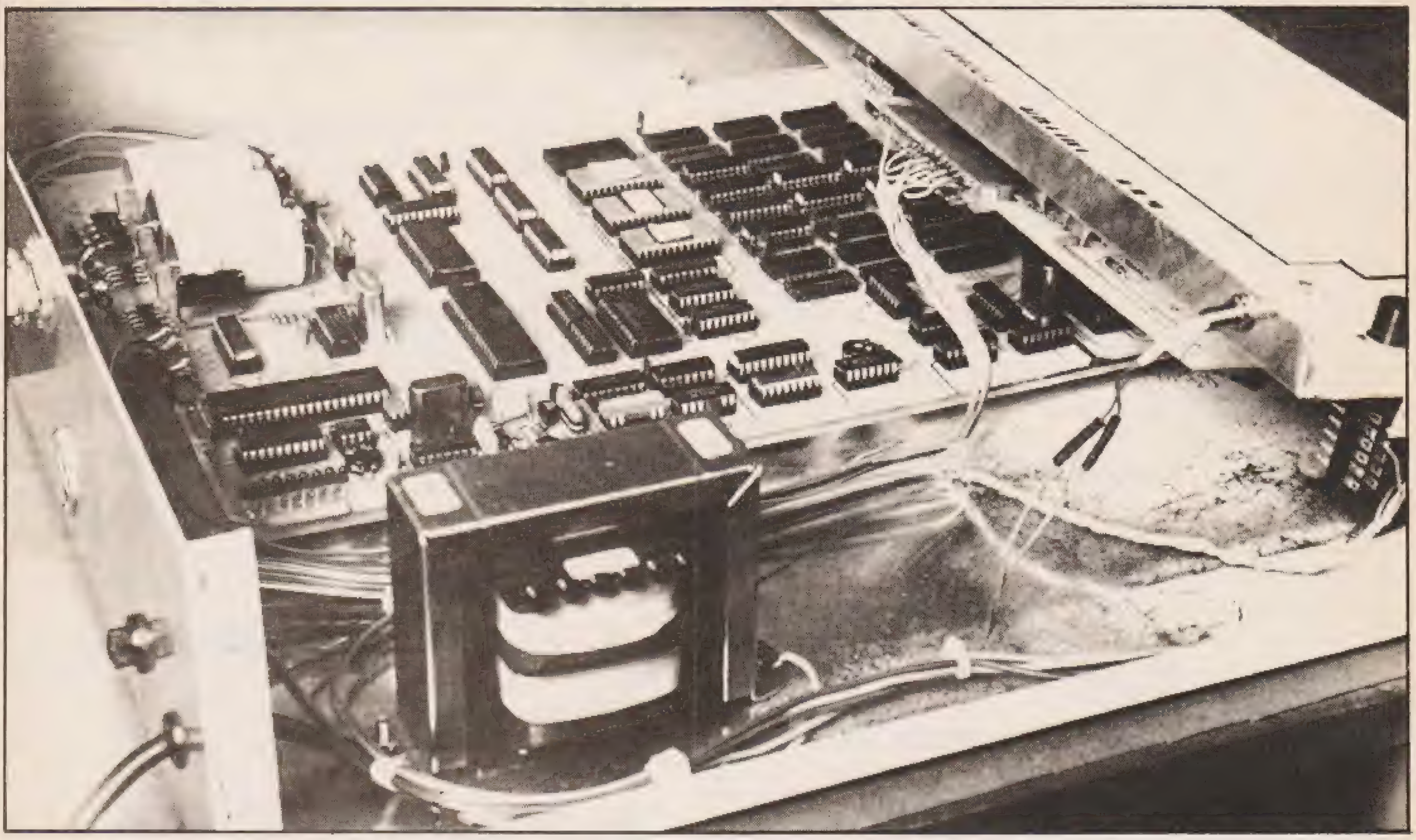
Veronis. MICROPROCESSOR **£12.85**

Zaks, R. INTRODUCTION TO PERSONAL AND BUSINESS COMPUTING **£8.50**

Zaks, R. MICROPROCESSORS FROM CHIPS TO SYSTEMS **£7.50**

Note that all prices include postage and packing. Please make cheques, etc, payable to Computing Today Book Service (Payment in U.K. currency only please) and send to:

Computing Today Book Service,
P.O. Box 79, Maidenhead, Berks.



Interface your Triton to a cheap printer unit with this feature. Much more besides

The program described in this article, with its simple interface, will allow you to use a surplus GPO teleprinter on a TRITON. These machines, the author used a CREED 7B, can be purchased relatively cheaply from electronic brokers and because of the typewriter style of output are suitable for letters. Another advantage is that they use cheap roll type paper, rather than continuous forms.

They do have drawbacks in that they are slow and noisy but this will probably not worry too many people on a tight budget. The program contains an ASCII to Baudot conversion routine, a miniscule letter editor and a Hex dump routine to print out any section of TRITONS memory. Flow charts of the program are given to make the listing understandable.

Typeright Operation

To use the Mini Editor simply hit G 1602. The cursor will now step to a new line. You may now enter a page, or pages, of text up to an amount limited by the available on-board RAM. With a full set of RAM you may store up to 2000 characters but this includes spaces etc. To create a new line you must use RETURN and then line feed (Roll screen or ESC) as simply going over the edge of the screen will not create a new printer line.

To correct errors within a line Control H will backspace over characters, but do not try to backspace over lines other than the current one as this may look OK on the screen but not to the RAM. Other cursor controls should NOT be used as they are not handled by the program.

Any characters which are not recognised, or are not contained in the Baudot code are printed as a space. When you have finished composing your text on the screen, don't be afraid to overtype any scrolled lines, hit Control D. The message 'HOW MANY? 1 TO 9' will be displayed and the input of any number in that range will start the printer and output the required number of copies. After printing is completed you are asked if you want more, if not the screen will be cleaned for more text input.

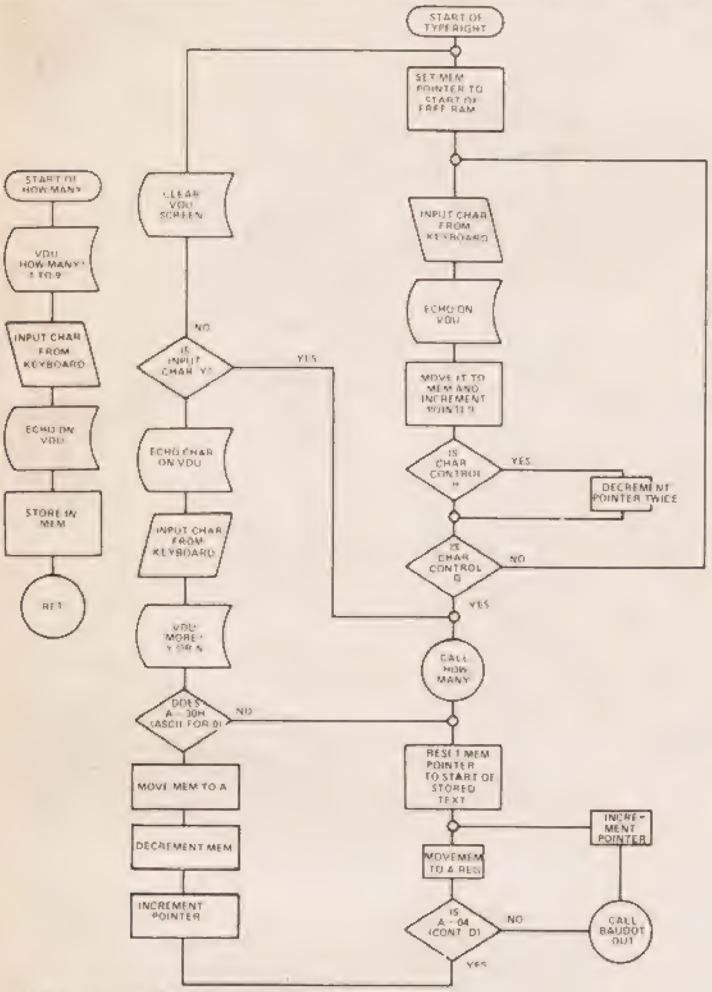
Further Facilities

To use the Hex dump facility hit P 1681 and enter the starting address of the section you want dumped, low order byte first. Keying in G 1680 will cause the printer to output a line of Hex code until it finds an address ending in 0H, where upon it executes a carriage return and stops. Hitting any key at this point will cause the next line of 16 bytes to be output.

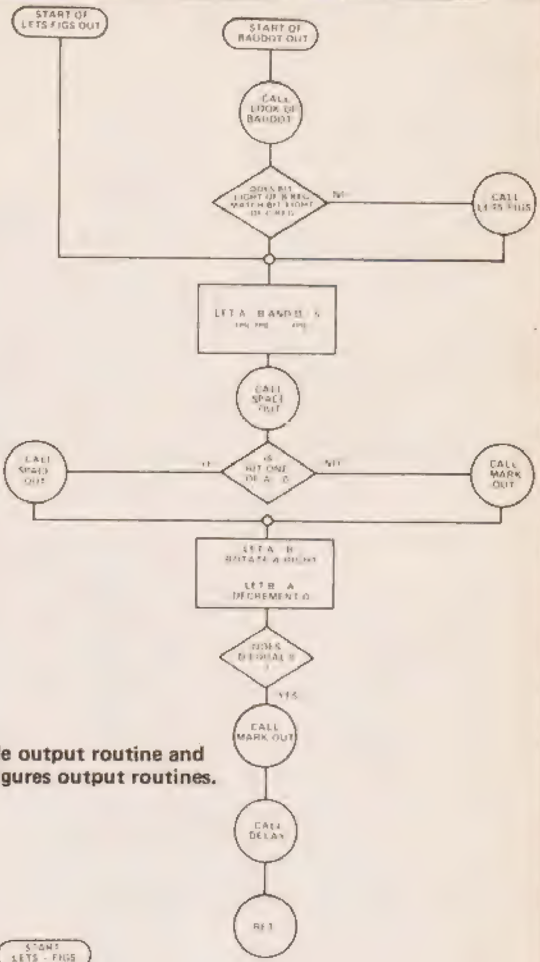
The Baudot code conversion may be used on its own with a program written between 1600 and 16E3, calling 16E3 will cause the accumulator contents to be dumped. It should be noted that the teleprinters being used always print one letter behind the code received as they use a mechanical decode, hence you should always leave a space or line feed at the end.

Cost And Implementation

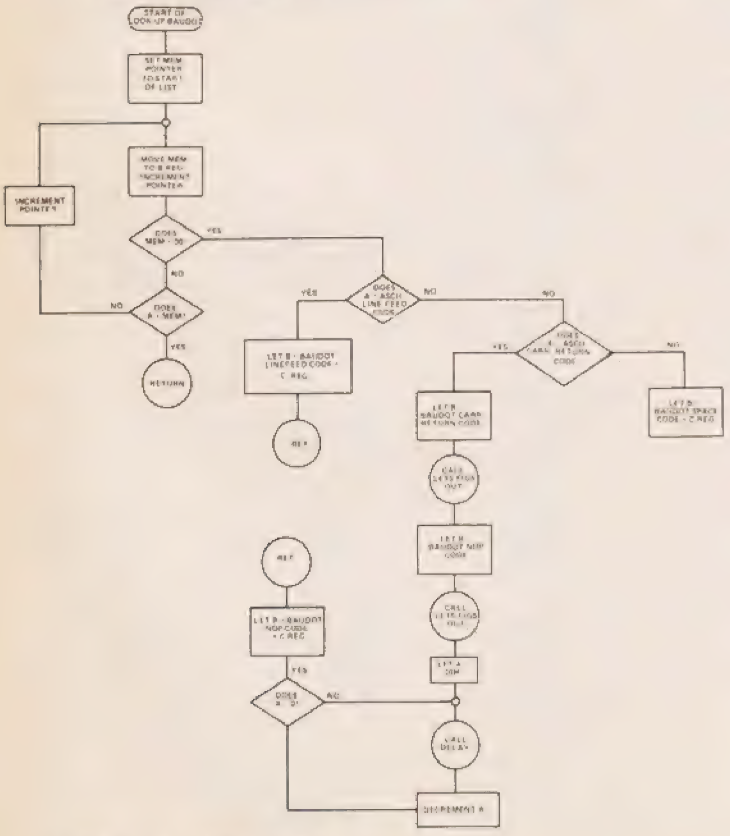
The printer that the author used cost £20 and took a fair amount of work to become operational, however you can spend a little more and buy one that runs off 240V, his uses two 12V car batteries. You should try to extract a promise from the dealer to make good any missing bits (pun not intended) as it is difficult to ensure that all the mechanics are present. The interface circuitry and connections are shown in Fig 1.



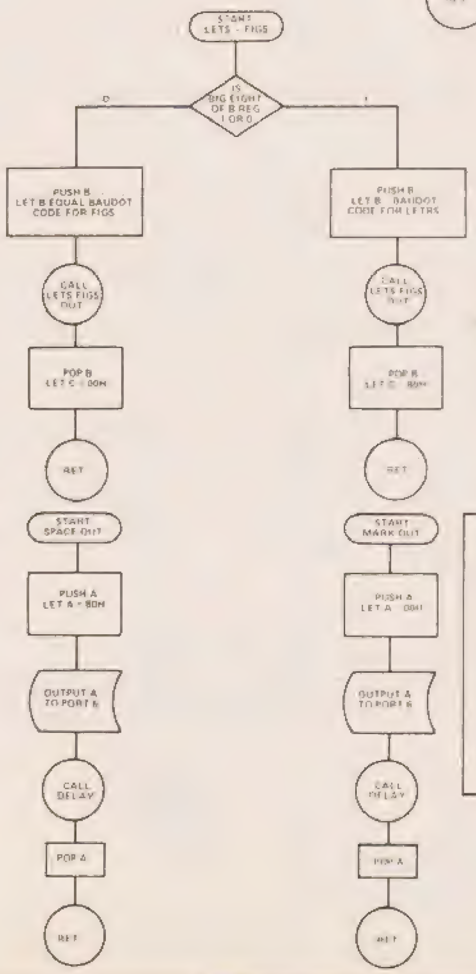
The main routine flowchart.



The Baudot Code output routine and the Letters or Figures output routines.

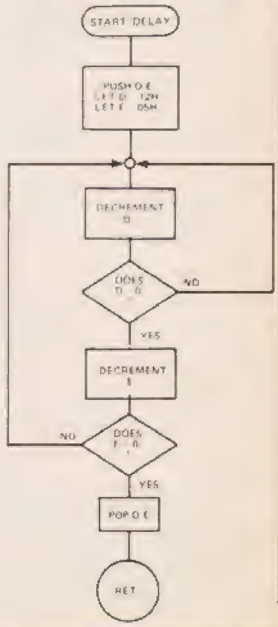


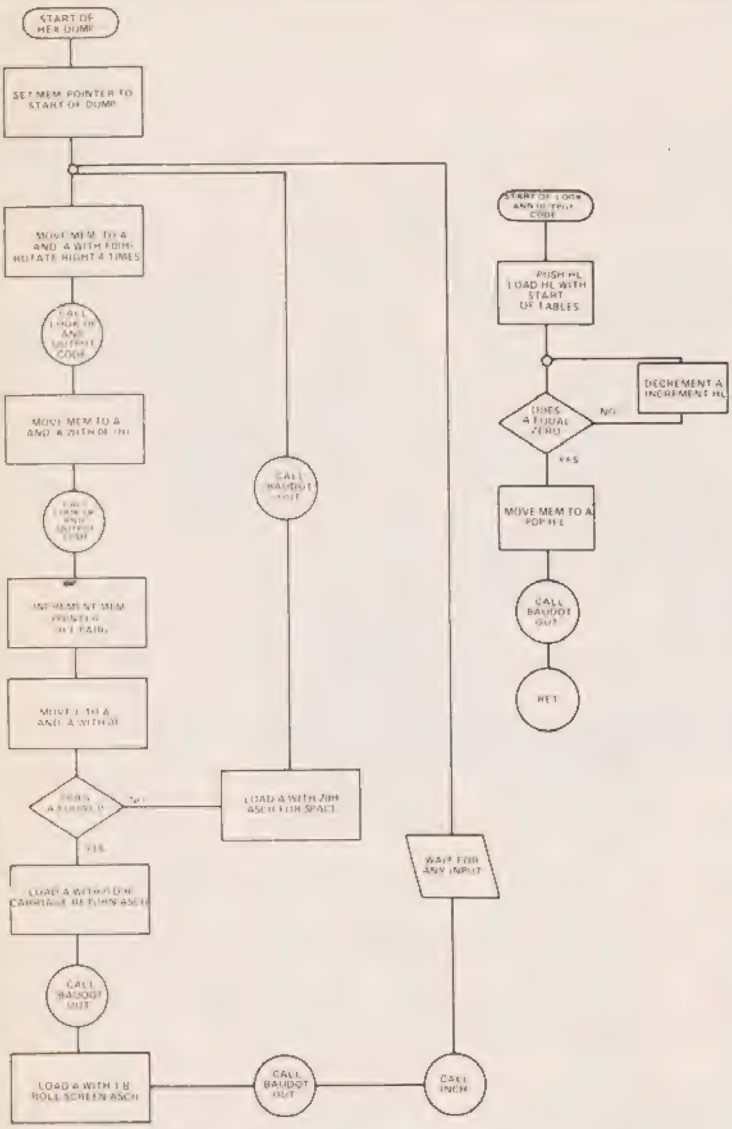
The Baudot look-up routine.



The output code check routines.

The delay subroutine.





The Hex dump routine.

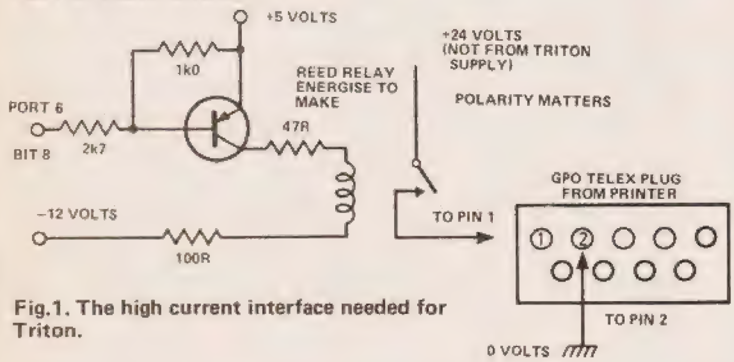
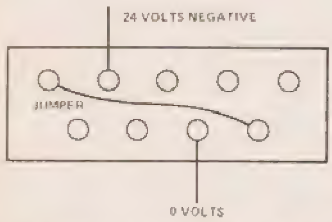
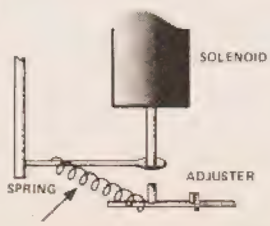


Fig.1. The high current interface needed for Triton.



To use the Teleprinters own keyboard to trouble-shoot the printer and set up the program you may connect the signal plug as shown.



If the printer's solenoid spring has been disconnected you should re-connect it so that the solenoid arm is moved to the right when it is NOT energised. The arm may now be adjusted for proper operation.

```
1600 06   END ADDRESS FOR TAPING
1601 18
1602 21   LOAD HL PAIR
1603 06   START OF FREE RAM
1604 18
1605 CD   CALL ECHOCH
1606 3B
1607 00
1608 77   MOVE ACC. TO MEM.
1609 23   INCREMENT HL
160A FE   COMPARE ACC. TO-
160B 08   CODE FOR BACKSPACE
160C C2   JUMP IF NOT
160D 12
160E 16
160F 2B   DECREMENT HL

1610 2B   AND AGAIN
1611 00   NOP
1612 FE   COMPARE TO-
1613 04   EOT CODE
1614 C2   JUMP IF NOT
1615 05
1616 16
1617 CD   CALL SUBR. 'HOW MANY'
1618 30
1619 16
161A 21   RELOAD START IN HL
161B 06
161C 18
161D 7E   MOVE MEM. TO ACC.
161E FE   COMPARE TO-
161F 04   EOT

1620 CA   JUMP IF MATCH
1621 3E   TO SUBR. REPEAT PRINTOUT
1622 16
1623 CD   CALL 'BAUDOT OUT'
1624 E3
1625 16
1626 23   INCREMENT HL
1627 C3   JUMP BACK FOR NEXT LETTER
1628 1D
1629 16
162A 00   NOP
162B 00
162C 00
162D 00
162E 00
162F 00
1630 11   LOAD DE PAIR
1631 60   START ADDRESS
1632 16     OF STRING
1633 CD   CALL PSTRNG
1634 2B
1635 00
1636 CD   CALL ECHOCH
1637 3B
1638 00
1639 77   MOVE ACC. TO MEM.
163A C9   RETURN
163B 00
163C 00
163D 00
163E 23   INCREMENT HL
163F 35   DECREMENT MEM.

1640 7E   MOVE MEM. TO ACC.
1641 FE   COMPARE TO-
1642 30     CODE FOR ZERO
1643 C2   JUMP IF NOT
1644 1A     TO SUBR.
1645 16     'START PRINTOUT'
1646 11   LOAD DE PAIR
1647 72     START ADDRESS
1648 16     OF STRING
1649 CD   CALL PSTRNG
164A 2B
```

```
164B 00
164C CD     CALL ECHOCH
164D 3B
164E 00
164F FE     COMPARE TO-

1650 59     CODE FOR Y
1651 CA     JUMP IF MATCH
1652 17       TO SUBR.
1653 16       'HOW MANY?'
1654 3E     LOAD ACC. WITH-
1655 0C     CODE FOR CONT. L
1656 CD     CALL OUTCH
1657 13
1658 00
1659 C3     JUMP BACK
165A 02       TO START
165B 16
165C 00
165D 00
165E 00
165F 00
1660 48     START OF STRING
1661 4F       'HOW MANY'
1662 57
1663 20
1664 4D
1665 41
1666 4E
1667 59
1668 3F
1669 20
166A 31
166B 20
166C 54
166D 4F
166E 20
166F 39
1670 20
1671 04
1672 4D     START OF STRING
1673 4F      'MORE?'
1674 52
1675 45
1676 3F
1677 20
1678 59
1679 20
167A 4F
167B 52
167C 20
167D 4E
167E 20
167F 04     EOT
1680 21     LOAD HL PAIR
1681 --     ENTER ADDRESS FOR START
1682 --       OF HEX DUMP
1683 7E     MOVE MEM. TO ACC.
1684 E6     AND ACC. WITH-
1685 F0
1686 0F     ROTATE ACC.RIGHT
1687 0F       FOUR TIMES
1688 0F
1689 0F
168A CD     CALL SUBROUTINE
168B B7       'LOOK UP AND
168C 16        OUTPUT CODE'
168D 7E     MOVE MEM. TO ACC.
168E E6     AND ACC. WITH-
168F 0F
1690 CD     CALL SUBR.
1691 B7       'LOOK UP AND
1692 16       OUTPUT CODE'
1693 23     INCREMENT HL
1694 7D     MOVE L TO ACC.
1695 E6     AND ACC. WITH-
```

```
1696 0F
1697 FE     COMPARE TO-
1698 00
1699 CA     JUMP IF MATCH
169A A4       TO SUBR.
169B 16       'END LINE'
169C 3E      LOAD ACC. WITH-
169D 20        CODE FOR SPACE
169E CD     CALL 'BAUDOT OUT'
169F E3

16A0 16
16A1 C3     JUMP BACK FOR
16A2 83       NEXT BYTE
16A3 16
16A4 3E     LOAD ACC. WITH-
16A5 0D       CODE FOR CARR.RETURN
16A6 CD     CALL 'BAUDOT OUT'
16A7 E3
16A8 16
16A9 3E     LOAD ACC. WITH-
16AA 1B       CODE FOR ROLL SCREEN
16AB CD     CALL 'BAUDOT OUT'
16AC E3
16AD 16
16AE CD     CALL INCH
16AF 0B

16B0 00
16B1 C3     JUMP BACK TO
16B2 83       PRINT NEXT LINE
16B3 16
16B4 00
16B5 00
16B6 00
16B7 E5     PUSH HL
16B8 21     LOAD HL WITH
16B9 D0       START OF TABLE
16BA 16
16BB FE     COMPARE ACC. WITH-
16BC 00       ZERO
16BD CA     JUMP IF MATCH TO
16BE C5       SUBR. 'OUTPUT CODE'
16BF 16

16C0 3D     DECREMENT ACC.
16C1 23     INCREMENT HL
16C2 C3     JUMP BACK TO
16C3 BB       COMPARE TO ZERO
16C4 16
16C5 7E     MOVE MEM. TO ACC.
16C6 E1     POP HL
16C7 CD     CALL 'BAUDOT OUT'
16C8 E3
16C9 16
16CA C9     RETURN
16CB 00
16CC 00
16CD 00
16CE 00
16CF 00

16D0 30     START OF LOOK UP
16D1 31       TABLE FOR 'HEX DUMP'
16D2 32
16D3 33
16D4 34
16D5 35
16D6 36
16D7 37
16D8 38
16D9 39
16DA 41
16DB 42
16DC 43
16DD 44
16DE 45
16DF 46
```

```
16E0 00
16E1 30
16E2 00
16E3 CD     CALL 'LOOK UP
16E4 52       BAUDOT'
16E5 17
16E6 78     MOVE B TO ACC.
16E7 E6     AND ACC. WITH-
16E8 80       80H
16E9 B9     COMPARE WITH- C
16EA C4     CALL IF NO MATCH
16EB 35       'LETS-FIGS'
16EC 17
16ED D5     PUSH DE
16EE 1E     LOAD E WITH-
16EF 05       05H

16F0 CD     CALL 'SPACE OUT'
16F1 2A
16F2 17
16F3 78     MOVE B TO ACC.
16F4 E6     AND WITH-
16F5 01       01H
16F6 CC     CALL IF ZERO
16F7 2A       'SPACE OUT'
16F8 17
16F9 C4     CALL IF NOT ZERO
16FA 22       'MARK OUT'
16FB 17
16FC 1D     DECREMENT E
16FD CA     JUMP IF ZERO
16FE 06       'CHARACTER ENDS'
16FF 17

1700 78     MOVE B TO ACC.
1701 1F     ROTATE RIGHT
1702 47     MOVE ACC. TO B
1703 C3     JUMP BACK FOR
1704 F4       NEXT BIT
1705 16
1706 CD     CALL 'MARK OUT'
1707 22
1708 17
1709 CD     CALL 'DELAY'
170A 11
170B 17
170C D1     POP DE
170D C9     RETURN
170E 00
170F 00

1710 00
1711 D5     PUSH DE
1712 11     LOAD DE PAIR
1713 05
1714 12
1715 15     DECRMENT D
1716 C2     JUMP IF NOT ZERO
1717 15
1718 17
1719 1D     DECREMENT E
171A C2     JUMP IF NOT ZERO
171B 15
171C 17
171D D1     POP DE
171E C9     RETURN
171F 00

1720 00
1721 00
1722 3E     LOAD A WITH-
1723 00       00H
1724 D3     OUTPUT A TO PORT-
1725 06       06H
1726 CD     CALL 'DLAY'
1727 11
1728 17
1729 C9     RETURN
```

```
172A 3E     LOAD A WITH-
172B 80       80H
172C D3     OUTPUT A TO PORT-
172D 06       06H
172E CD     CALL 'DELAY'
172F 11

1730 17
1731 C9     RETURN
1732 00
1733 00
1734 00
1735 FE     COMPARE A TO-
1736 80       80H
1737 CA     JUMP IF MATCH TO-
1738 45       SUBR. 'LETS'
1739 17
173A 00     NOP
173B C5     PUSH B
173C 06     LOAD B WITH-
173D 1B     BAUDOT FOR FIGS
173E CD     CALL 'LETS-FIGS OUT'
173F ED

7740 16
1741 C1     POP B
1742 0E     LOAD C WITH-
1743 00       00H
1744 C9     RETURN
1745 C5     PUSH B
1746 06     LOAD B WITH-
1747 1F     BAUDOT FOR LETS
1748 CD     CALL 'LETS-FIGS OUT'
1749 ED
174A 16
174B C1     POP B
174C 0E     LOAD C WITH-
174D 80       80H
174E C9     RETURN
174F 00

1750 00
1751 00
1752 E5     PUSH HL
1753 21     LOAD HL WITH-
1754 9E       START OF
1755 17       BAUDOT TABLES
1756 46     MOVE MEM. TO B
1757 23     INCREMENT HL
1758 BE     COMPARE A TO MEM.
1759 23     INCREMENT HL
175A C2     JUMP IF NO MATCH TO-
175B 5F       SKIP RETURN
175C 17
175D E1     POP HL
175E C8     RETURN IF MATCH
175F F5     PUSH PSW

1760 7E     MOVE MEM. TO A
1761 FE     COMPARE TO-
1762 00       00H
1763 CA     JUMP IF MATCH TO-
1764 6A       SUBR. 'NOT
1765 17       IN TABLE'
1766 F1     POP PSW
1767 C3     JUMP BACK TO LOOK
1768 56       AT NEXT LOCATION
1769 17       IN TABLE
176A F1     POP PSW
176B E1     POP HL
176C FE     COMPARE A TO-
176D 1B       ROLL SCREEN CODE
176E F5     PUSH PSW
176F 79     MOVE C TO A

1770 C6     ADD TO A-
1771 02     BAUDOT FOR LINE FEED
1772 47     MOVE A TO B
1773 F1     POP PSW
1774 C8     RETURN IF ROLL SCREEN
1775 FE     COMPARE A TO-
1776 0D     CODE FOR CARR. RETURN
1777 C2     JUMP IF NOT TO-
1778 96       SUBR.'PRINT SPACE'
1779 17
177A 00
177B 00
177C 00
177D 3E     LOAD A WITH-
177E 08     BAUDOT FOR CARR.RETURN
177F 81     ADD C

1780 47     MOVE A TO B
1781 CD     CALL 'LETS-FIGS OUT'
1782 ED
1783 16
1784 41     MOVE C TO B
1785 CD     CALL 'LETS-FIGS OUT'
1786 ED
1787 16
1788 3E     LOAD A WITH-
1789 10       10H
178A CD     CALL 'DELAY'
178B 11
178C 17
178D 3D     DECRMENT A
178E C2     JUMP IF NOT ZERO
178F 8A       BACK FOR MORE DELAY

1790 17
1791 41     MOVE C TO B
1792 C9     RETURN
1793 00
1794 00
1795 00
1796 79     MOVE C TO A
1797 C6     ADD TO A-
1798 04       BAUDOT FOR SPACE
1799 47     MOVE A TO B
179A C9     RETURN
179B 00
179C 00
179D 00
179E 83     START OF LOOK UP
179F 41       TABLE, BAUDOT CODE

17A0 99       IN FIRST BYTE,
17A1 42       ASCII CODE IN NEXT
17A2 8E       AND SO ON.
17A3 43     BAUDOT LETTERS AND
17A4 89       FIGURES ARE
17A5 44       DIFFERENCIATED
17A6 81       BY ADDING 80H
17A7 45       TO THE LETTERS.
17A8 8D
17A9 46
17AA 9A
17AB 47
17AC 94
17AD 48
17AE 86
17AF 49

17B0 8B
17B1 4A
17B2 8F
17B3 4B
17B4 92
17B5 4C
17B6 9C
17B7 4D
17B8 8C
17B9 4E
17BA 98
17BB 4F
17BC 96
17BD 50
17BE 97
17BF 51

17C0 8A
17C1 52
17C2 85
17C3 53
17C4 90
17C5 54
17C6 87
17C7 55
17C8 9E
17C9 56
17CA 93
17CB 57
17CC 9D
17CD 58
17CE 95
17CF 59

17D0 91
17D1 5A
17D2 17
17D3 31
17D4 13
17D5 32
17D6 01
17D7 33
17D8 0A
17D9 34
17DA 10
17DB 35
17DC 15
17DD 36
17DE 07
17DF 37

17E0 06
17E1 38
17E2 18
17E3 39
17E4 16
17E5 30
17E6 19
17E7 3F
17E8 03
17E9 2D
17EA 0E
17EB 3A
17EC 0D
17ED 25
17EE 1A
17EF 40

17F0 14
17F1 24
17F2 0F
17F3 28
17F4 12
17F5 29
17F6 1C
17F7 2E
17F8 0C
17F9 2C
17FA 05
17FB 27
17FC 1E
17FD 3D
17FE 1D
17FF 2F

1800 11
1801 2B
1802 00     00 MUST BE IN
1803 00       THESE LOCATIONS
1804 00       TO RETURN
1805 00       FROM TABLES
```

TRITON TYPECAST

```
1806 --      START OF FREE RAM
               TEXT ENTERED FROM
               HERE ON


             'TYPERIGHT'

             START ADDRESSES FOR SUBROUTINES

1602     MINI-EDITOR (TO PRINT TEXT)
161A     START PRINTOUT
1630     HOW MANY?
1646     MORE?
1660     STRING 'HOW MANY'
1672     STRING 'MORE?'
1680     HEX DUMP (ENTER START IN 1681-1682)
16A4     END LINE
16B7     LOOK UP AND OUTPUT CODE
16C5     OUTPUT CODE
16D0     START OF LOOK UP TABLE FOR HEX DUMP
16E3     BAUDOT OUT (MAY BE USED WITHOUT ABOVE
           ROUTINES. CONVERTS CODE IN ACC. TO
           BAUDOT AND SENDS IT OUT PORT 06H, THEN
           RETURNS)
16ED     LETS-FIGS OUT (OUTPUTS CONTROL CODES)
1711     DELAY  (1713-1714 MAY BE ALTERED TO CHANGE
           TRANSMISSION SPEED)
1722     MARK OUT
172A     SPACE OUT
1735     LETS-FIGS
1752     LOOK UP BAUDOT
177D     SPECIAL ROUTINE FOR CARRAIGE RETURN
1789     MAY BE ALTERED FOR DIFFERENT DELAY
            AFTER CARRAIGE RETURN
1796     PRINT SPACE
179E     START OF LOOK UP BAUDOT TABLES
1806     START OF FREE RAM FOR TEXT STORAGE

                 EXAMPLE OF HEX DUMP FUNCTION
                 PROGRAM DUMPS ITSELF
                 (LOAD 1681-2 WITH
                 START ADDRESS 1600)


06 18 21 06 18 CD 3B 00 77 23 FE 08 C2 12 16 2B
2B 00 FE 04 C2 05 16 CD 30 16 21 06 18 7E FE 04
CA 3E 16 CD E3 16 23 C3 1D 16 00 00 00 00 00 00
11 60 16 CD 2B 00 CD 3B 00 77 C9 00 00 00 23 35
7E FE 30 C2 1A 16 11 72 16 CD 2B 00 CD 3B 00 FE
59 CA 17 16 3E 0C CD 13 00 C3 02 16 00 00 00 00
48 4F 57 20 4D 41 4E 59 3F 20 31 20 54 4F 20 39
20 04 4D 4F 52 45 3F 20 59 20 4F 52 20 4E 20 04
21 00 16 7E E6 F0 0F 0F 0F 0F CD B7 16 7E E6 0F
CD B7 16 23 7D E6 0F FE 00 CA A4 16 3E 20 CD E3
16 C3 83 16 3E 0D CD E3 16 3E 16 CD E3 16 CD 06
00 C3 83 16 00 00 00 E5 21 D0 16 FE 00 CA C5 16
3D 23 C3 BB 16 7E E1 CD E3 16 C9 00 00 00 00 00
30 31 32 33 34 35 36 37 38 39 41 42 43 44 45 46
00 00 00 CD 52 17 78 E6 80 B9 C4 35 17 D5 1E 05
CD 2A 17 78 E6 01 CC 2A 17 C4 22 17 1D CA 06 17
78 1F 47 C3 F4 16 CD 22 17 CD 11 17 D1 C9 00 00
00 D5 11 05 12 15 C2 15 17 1D C2 15 17 D1 C9 00
00 00 3E 00 D3 06 CD 11 17 C9 3E 80 D3 06 CD 11
17 C9 00 00 00 FE 80 CA 45 17 00 C5 06 16 CD ED
16 C1 0E 00 C9 C5 06 1F CD ED 16 C1 0E 80 C9 00
00 00 E5 21 9E 17 46 23 BE 23 C2 5F 17 E1 C8 F5
7E FE 00 CA 6A 17 F1 C3 56 17 F1 E1 FE 16 F5 79
C6 02 47 F1 C8 FE 0D C2 96 17 00 00 00 3E 08 81
47 CD ED 16 41 CD ED 16 3E 10 CD 11 17 3D C2 8A
17 41 C9 00 00 00 79 C6 04 47 C9 00 00 00 83 41
99 42 8E 43 89 44 81 45 8D 46 9A 47 94 48 86 49
8B 4A 8F 4B 92 4C 9C 4D 8C 4E 98 4F 96 50 97 51
8A 52 85 53 90 54 87 55 9E 56 93 57 9D 58 95 59
91 5A 17 31 13 32 01 33 0A 34 10 35 15 36 07 37
06 38 18 39 16 30 13 3F 03 2D 0E 3A 0D 25 1A 40
14 24 0F 28 12 29 1C 2E 0C 2C 05 27 1E 3D 1D 2F
11 26 00 00 00 00 20 20 20 20 20 20 20 20 20 20
```

Phil Cornes.

BEGINNING BASIC

This month we present the solution to the sort problem and take a look at Binary searches

Last month we left you with a program to solve about the workings of our two list merge program. As we are about to give you the answer to that problem, it would help if you had last month's article in front of you.

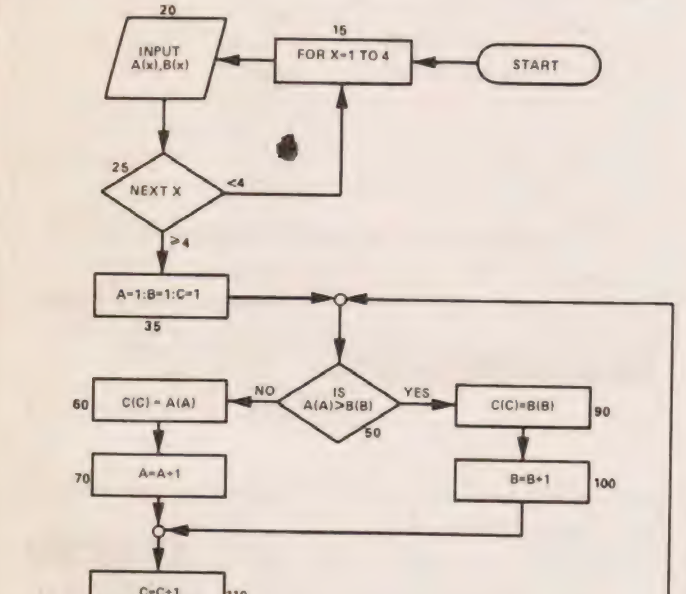


Fig.1. The flowchart for last months problem, the shaded area represents lines 120 to 210.

Sorting The Algorithm

The object of the problem was to try to decide what lines 120-210 of the two list merge program were included for. You may remember that the program was to take two lists of numbers, A(X) and B(X), each containing 4 items in ascending numerical order and produce from these a single list, C(X), of 8 items which was to have been the result of merging the two lists together in such a way that the resulting list was also in ascending order. A simple algorithm was given to solve the problem and you were expected to dry run the program listing produced from this algorithm. The best way to try to understand the operation of a program like this is to draw a near flowchart. (See Fig.1).

As far as possible, the flowchart boxes have been numbered to correspond with the relevant program line numbers given last month. The shaded area in Fig.1 corresponds to lines 120-210 of the program. The flowchart boxes labelled 15, 20 & 25 allow you to input the two number lists A(X) and B(X). Note that these must be in ascending numerical order. Flowchart box 35 sets up initial values for A, B and C which will be used as pointers in lists A(X), B(X) and C(X) respectively. Box 50 now compares the next two items in lists A(X) and B(X) using pointers A and B to see which of A(A) and B(B) is larger (to start off with, both A and B have the value 1 so we are comparing A(1) with B(1). If A(A) is larger than B(B) then we branch to box 90 and put the contents of B(B) into C(C). So here we make C(1) equal to B(1) (because B(1) was the smaller of A(1) or B(1)) and then we increment both pointers B and C by 1. A and B are now tested to see if either of them has reached 5. In this case they have not and so we branch back to box 50 to compare A(A) with the new value of B(B). (Here A(1) and B(2)). We will suppose this time that A(1) is smaller than B(2) (though, of course, this need not be the case) so we branch to box 60 and put A(A) into C(C) (Here A(1) into C(2)). We then execute boxes 70 and 110 thus incrementing A and C by one (so now A=2, B=2, C=3). A and B are now both tested again to see if either has reached 5 and as they have not, we branch back to box 50 to compare A(2) and B(2) etc. You should see that this process is repeated until one of the lists A(X) or B(X) has been exhausted so that either A or B has the value 5. When this occurs, we encounter the programs line 120-210 mentioned last month (shaded area of flowchart Fig.1) and the process changes slightly. In box 20, originally, we only entered 4 items in each list so that when A or B equals 5 (say A) we will try to compare the next item in list B with the non-existent value A(5) unless something happens to change the flow of the program. This is in fact done. When A=5 any items that are left in list B must all be larger than any of the items that were in list A and they must also be in ascending numerical order so that all that is required to complete list C is to transfer the remaining items in list B directly onto the end of list C without any comparisons being necessary. This is done in boxes 150-170 (If B=5 and A 5 then the same argument applies as before and the transfer of the last items of list A to the end of list C is dealt with by flowchart boxes 190-210).

The last few flowchart boxes form a FOR NEXT loop which is used to print out the merged list C.

A Simpler Solution

The last part of last month's homework asked you to see if you could think of a way of simplifying the program by replacing lines 120-210 with something else. The program listing given below is one possible solution — I leave you to work out how it operates.

```
 10 DIM A(5), B(5), C(8)
 15 FOR X=1TO4
 20 INPUT A(X), B(X)
 25 NEXT X
 30 A=1
 35 B=1
 40 C=1
 43 A(5)=1E30
 47 B(5)=1E30
 50 IF A(A) > B(B) THEN 90
 60 C(C)=A(A)
 70 A=A+1
 80 GOTO 110
 90 C(C)=B(B)
100 B=B+1
110 C=C+1
120 IF C=9 THEN 220
140 GOTO 50
220 FOR C=1TO8
230 PRINT C(C)
240 NEXT C
250 END
```

NOTE For those of you who have not met scientific representations of numbers before, 1E30 in lines 43 and 47 means 1x10^{30} or 1 with 30 zeros after it like this:1000000000000000000000000000000. As you can see, it's quite a large number and much easier to write as 1E30.

Simple Sort

So far we have examined a simple search algorithm and a merge algorithm, before we go on to consider anything more complex in either of these directions, we will take a look at a simple sorting algorithm and program.

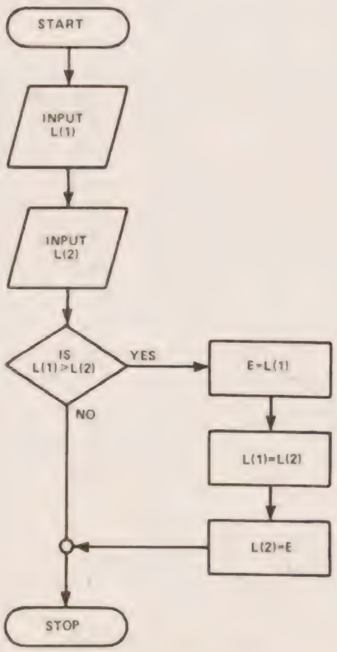


Fig.2. Simple sort algorithm for increasing order of 2 items.

Imagine a list containing two items L(1) and L(2) and now imagine an algorithm to place these two items in ascending numerical order — easy! I hear you say, and so it is. (see Fig.2) But now imagine this list increased to contain 3 items L(1)—L(3). Can we still sort them? Yes, obviously we can and in very similar fashion (see Fig.3).

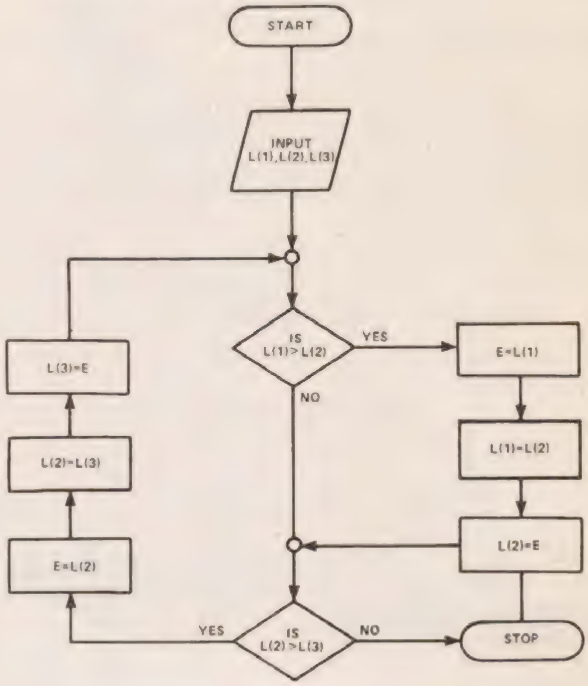


Fig.3. Sorting with three items.

You should begin to see a general algorithm appearing from all of this which can be used for any length of list (N). For P=1 to N, compare L(P) with L(P+1) and reverse these if necessary, then go on to the next value of P. This FOR NEXT loop should be repeated until we manage to go once completely through the FOR NEXT loop without having to reverse any pair of numbers L(P) and L(P+1). See Fig.4 and the following program.

```
 10 INPUT N
 20 FOR P=1 TO N
 30 INPUT L(P)
 40 NEXT P
 50 F=0
 60 FOR P=1 TO N–1
 70 IF L(P) > L(P+1) THEN 110
 80 NEXT P
 90 IF F=1 THEN 50
100 END
110 F=1
120 E=L(P)
130 L(P)=L(P+1)
140 L(P+1)=E
150 GOTO 80
160 END
```

I leave you to work out the details of the operation of this program. One unusual point you will notice is that part of the work of a FOR NEXT loop in this program is carried on in a part of the program that is physically outside the loop. This is perfectly acceptable even though we have not met it before.

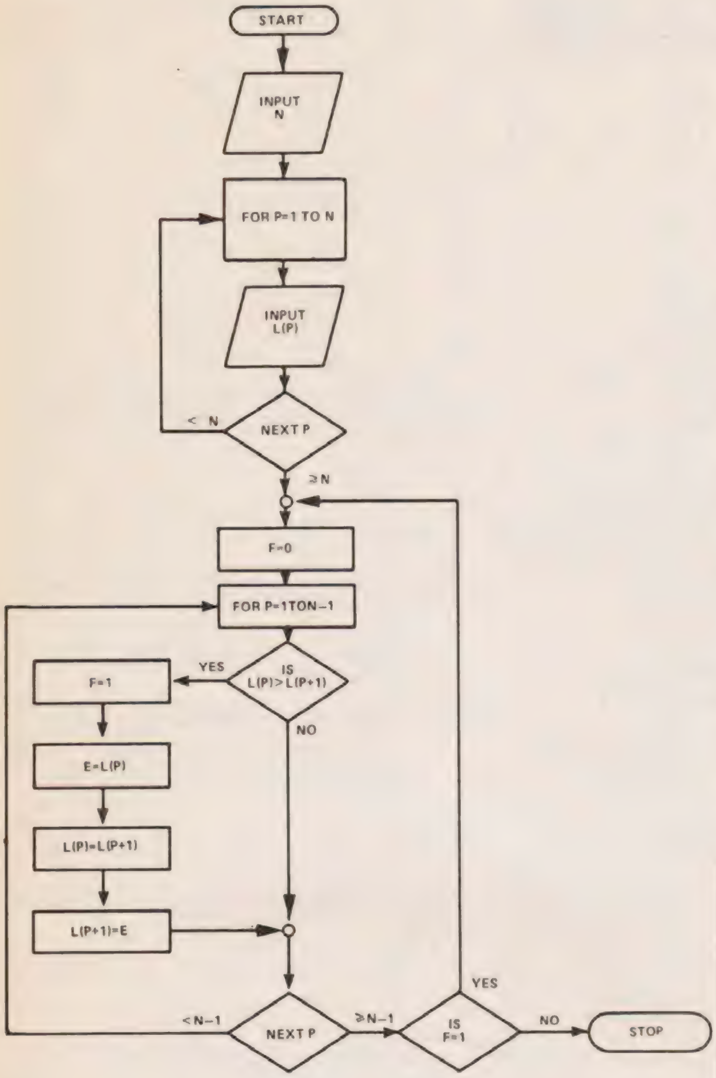


Fig.4. Flowchart for the program to perform a simple sort on any number of entries.

The Binary Search

Now we have a sort algorithm of sorts (ouch!) there is a far more efficient search algorithm that we can use that depends for its operation upon the fact that the list of items we are to search are in numerical order. This is called "The Binary Search" for reasons that will become more obvious as we describe it. If we take any list of numbers in any numerical order and try to find one particular number in the list, then if we divide the list into two equal halves, we can say for sure that if the number exists at all in the list it must reside in one of the two halves, though we would not know which. This may seem very obvious, but if we now specify that the list is numerically ordered, then we can say which half of the list contains our chosen number simply by comparing the contents of the halfway location with our chosen numbers. If these contents are larger than our number then our number must be contained in the lower half of the list and vice versa. Having eliminated half the list, we can now consider the remaining half as a complete list and split this into two and test again, and so on until we either find the item we are searching for, or prove it not to be contained in the list.

This may sound a little complex when compared with the previous search algorithm but when we look at how much more efficient this algorithm is than the previous one, we will certainly see that for any reasonable length list the extra complexity is more than justified.

Before we can compare the efficiency of the two algorithms we must find a criterion that we can reasonably consider as a measure of efficiency for a search algorithm. Such a criterion may, for example, be measured as follows:— Every time we look at one of the locations in the list and compare it's contents with our chosen number we will call this operation one comparison. Our criterion will then be the number of comparisons we have to make before we find our chosen number or prove it is not contained within the list. We will make this measure less random by ensuring that our chosen number is in the worst possible position as far as location is concerned and we should now have quite a good test. For a list of say 1000 items our first algorithm could take up to 1000 comparisons whereas the binary search can not take more than 10 or 11 comparisons — almost 100 times as fast.

Try This Yourself

For homework this month you might try to draw a flowchart to illustrate the operation of the binary search. Next month we look at a very efficient sort algorithm which makes our binary search even more attractive.

MINI-ADS & CLASSIFIED

CLASSIFIED INFORMATION

Semi-Display:—
1- 3 insertions — £5.00 per single column centimetre
4-11 insertions — £4.50 per s.c.c.
12 insertions — £4.00 per s.c.c.

Classified:—
19 pence per word (minimum 25 words)
Box number £1.00 extra

ALL ADVERTISEMENTS IN THIS SECTION MUST BE PRE-PAID
Closing date:— 2nd Friday in month preceding publication
Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request)
Cheques and postal orders should be crossed and made payable to 'Computing Today'

CLASSIFIED ADS, COMPUTING TODAY, 145 CHARING CROSS ROAD, LONDON WC2H 0EE (Tel. 01-437 1002)

THE NEW uHEX 480 EPROM PROGRAMMER 2704 and 2708

Control programs for Z80, 8080, 6800, 6500. Please state machine.
Programs permit programming any length block into the eprom, so even unexpanded machines can now program eproms.
Needs only +5V, +12V, −5V.
Host computers require a PIO (PIA) complete kit only £35 or ready built and tested £5 extra.

PIO, PIA INTERFACE PANELS
Available for Z80/8080, 6800/6500

THE uHEX 416 DELUXE EPROM PROGRAMMER
Push Button Selection for 2704, 2708, 2716. Still only £65.

All prices inclusive. SAE for further information about these products.

MICRO HEX COMPUTERS
2 Studley Rise, Trowbridge, Wilts.

INTENSIVE WEEKEND COURSES IN BASIC

including hands-on mini computer operation.

This short intensive course is intended to instruct from minimal knowledge to an operational capability of computer programming in BASIC high level language.
The course is fully residential from Friday evening to Sunday afternoon.
Option of non-residential weekend, weekday evening and weekday courses available if required.

For further details of dates available, fees, etc. **Phone (0401) 43139, or write to:**
Dept CT
CLEVELAND BUSINESS SERVICES
Cleveland House, ROUTH
Beverley, North Humberside

CHIPTESTER FOR NASCOM 1

Brand new fantastic product for Nascom 1 owners.
Use your Nascom to test all those surplus and suspect I.C.'s and sort out the good ones from the duff ones.
Just clip the I.C. in to the test clip provided and use the Nascom to examine the logic levels on each input and output. A graphical representation of the chip showing all pins and their logic levels is shown on the screen.
Tests all 5v TTL I.C.'s up to 16 pins. Includes commands for automatic and manual testing of logic levels of all inputs and outputs simultaneously.
All inputs may be changed under keyboard or program control including continuous test mode.
Propagation delays may be calculated using an oscilloscope.
Routines to test I.C.'s may be stored on cassette for later use.
Uses 2 ports on minimum Nascom system. No expansion whatever required.
Comes complete with all parts, full comprehensive instruction book, cassette with program and sample programs for popular I.C.'s.
Send S.A.E. for free descriptive leaflet.
Price inclusive of VAT and postage **£19.50**
Orders will be dealt with by return of post subject to availability of stock.
Access and Barclaycard welcome.
Stockists of Apple Microprocessors and Peripherals.

STRATHAND
44 St. Andrews Square,
Glasgow G1 5PL
041-552 6731

Tel. order welcome with Access and Barclaycard.
Now on Telex 777268, 24 hours service

50 Hz SUPERBOARDS BRITISH MODEL

from **£190** + VAT
Fully built tested and set up
Authorised dealer backup
C.T.S. 1 Higher Calderbrook, Littleborough, Lancs. Tel. Littleborough 79332 anytime

JADE SSM S100 INTER SYSTEMS WMC

S100 Motherboards — 9 slot and 12 slot
CPU Boards Z80 and Z80A kit or A & T
I/O Boards with tape interface kit or A & T
8K 450ns Static Ram Kit or A & T
16K 450ns Static Ram Kit or A & T
Video Boards with fine graphics
Floppy Disc controller D/S D/D
Main frame cabinets with Motherboard and PSU
5.25" and 8" Floppy Disc's Eprom Boards

S.A.E. for price list and special offers

New address:
MBE (Electronic Specialists)
508-516, Alum Rock Road,
Birmingham, B8 3HX.
Tel. 021-327 1497/6313

COLOUR MODULATOR

Kit
£9.95
inc UHF Modulator

FOR ALL TV GRAPHICS!
Red Green Blue inputs (can be mixed)
Coming soon Nascom Colour Kit
£39.95 inc Graphics Software

WILLIAM STUART SYSTEMS
Dower House, Billericay Road, Herongate, Brentwood, Essex
CM13 3SD Tel (0277) 810244
Barclaycard, Access welcome

GAMES FOR NASCOM

Five new Nascom games

SNARK HUNT, CHASE AND TRAIL, BULLS AND COWS, FOUR IN A ROW AND LIFE ON CASSETTE.

All run on minimum Nascom system. All five recorded both sides. Fantastic value complete with printed instruction booklet. £11.50 inc. VAT and postage and packaging.

(Instruction Booklet only £1.00, refunded if games cassette purchased).

STRATHAND
44 St. Andrews Square,
Glasgow G1 5PL
041-552 6731

Tel. order welcome with Access and Barclaycard
Now on Telex 777268, 24 hours service

TRITON FOR SALE

complete with Motherboard, Levels 5 and 6 BASIC, TRAP, 12K RAM,
Cost around £800
For sale for £700

Phone Steve on 240 0568
(10 am — 10 pm)

BIG GAMES for small systems written in Tiny Basic, also games tapes for Triton using L41 Basic, all run in under 3K. Send S.A.E. for list to B. Ewan 31 Chester Avenue, Bolton BL3 1XE.

STATIC RAM BARGAIN. New National Semiconductor MM2102AN-4L low power 1K x 1, 450ns. Price 75p each, inclusive. A. C. Gidlow, 7 Cypress Road, Newport, Isle of Wight. PO30 1EY.

TRITON MINI-COMPUTER, full 4K RAM. Professionally built and tested, several free programs including calibration tape. Buying a flat, therefore only £250. Ring 01-622 1952 daytime.

AD INDEX

TRS 80 Level II 16K Microcomputer with cassette player, library 100, N17, and snakes EEGs, only £400. Tel. 902-4198.

BEST VALUE SOFTWARE for PET. Multi-programme cassettes. Personal, educational, programming tutor, and games. SAE lists. Ridley Photo/Electronics Ltd., (SS1), Wallsend House, Pevensey Bay, Sussex.

ALTAIR 8800b complete with ¼ Mbyte disc, bootstrap, 2 Serial Ports, 32K, Microsoft BASIC, full documentation. Ideal business system. £980. Tel. Reading 861280 (evening) 596842 (day).

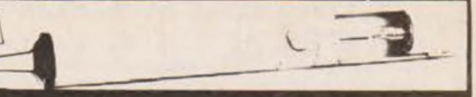
STACKS OF MEMORY! 4K Bit Static MOS Ram N.E.C. UPD 410 D-2 (100ns). £4.00 each plus 30p P & P & Data. P. Town Products, 30 Wolsey Drive, Walton-on-Thames, Surrey. KT12 3AZ. Tel: Walton 21078.

COMPUTER CASSETTE DUPLICATING, C15 — 33p each, inclusive + postage. All types duplicated, minimum 10. SAE Simon Stable Promotions, Inglenook, West End, Launton, Oxon. 08692 2831.

COMPUTERS, firmware, software. The best for less! Most makes bought, sold, exchanged, constructed. PET, ITT, ELF, New Sharp, Petsoft, Gemsoft, etc. Teletext decoder £215. Grandstand games computer £115. Sharp 10K £499 : 18K £599 : 22K £619 : 34K £699. Plus V.A.T/carriage. Phone: Optelco, Rayleigh 774089.

NASCOM 1, 16K RAM, 1200 Baud cassette interface, Comp. graphics, B-BUG, T.V. modified for direct Video. All in smart vero-case and working. £350. Also computer controlled dual cassette drive. £100, and Printer £20, not yet working. Tel: Scole 461 (evenings).

THE NEW

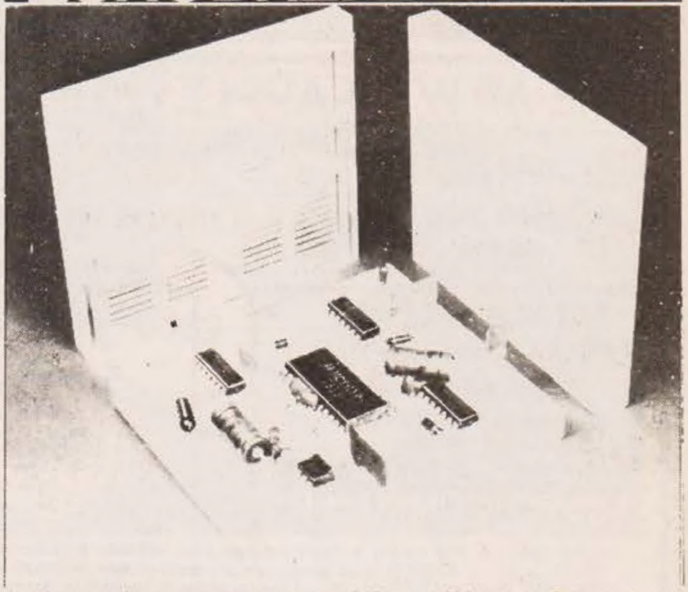


EUROSOLDERSUCKER

This 195mm long, all metal, high suction, desoldering tool with replaceable Teflon tip enables removal of molten solder from all sizes of pcb pads. Primed and released by thumb, it incorporates an anti recoil system and built in safety guard. **Only £7.25 inc. VAT & P.P.**

THE UNIQUE

EUROBREADBOARD



Logically laid out to accept both 0.3" and 0.6" pitch DIL packages as well as Capacitors, Resistors, LED's, Transistors and components with leads up to .85mm dia.

500 individual connections in the central breadboarding area, spaced to accept all sizes of DIL package without running out of connection points, plus 4 Integral Power Bus Strips around all edges for minimum inter-connection lengths.

All connection rows and columns are now numbered or lettered enabling exact location indexing.

Double-sided nickel silver contacts for long life (10K insertions) and low contact resistance (< 10m. ohms).

Easily removable, non-slip rubber backing allows damaged contacts to be rapidly replaced.

No other breadboard has as many individual contacts, offers all these features and costs only £6.20 each or £11.70 for 2 — inclusive of VAT and P.P.

Snip out and Post

David George Sales, r/o 74 Crayford High St., Crayford, Kent DA1 4EF

CT 11 79

David George Sales
r/o 74 Crayford High Street,
Crayford, Kent, DA1 4EF.

Please send me	1 EuroSolderSucker @ £7.25	☐	Please Tick
or	1 EuroBreadBoard @ £6.20	☐	
or	2 EuroBreadBoards @ £11.70	☐	

(All prices are applicable from July 1st, 1979 and include VAT & P.P., but add 15% for overseas orders).

Name ....

Company ....

Address ....

....

....

Tel. No ....

Please make cheque/P.O.'s payable to David George Sales

SEND LARGE S.A.E. FOR OUR FREE '79 CATALOGUE

AS SEEN IN
P.E. AUGUST, SEPTEMBER
OCTOBER 1979

EUROPES FASTEST SELLING ONE BOARD COMPUTER — JUST CHECK THE SPEC'S.

COMPUKIT UK101

SAMPLE TAPE WITH EXTENDED MACHINE CODE MONITOR AND DISSASSEMBLER INCLUDED FREE

LOW COST SUPERBOARD IN KIT FORM

The Compukit UK101 has everything a one board 'superboard' should have.

- ★ Uses ultra-powerful 6502 microprocessor.
- ★ 50Hz Frame refresh for steady clear picture (U.S.A. products with 60Hz frame refresh always results in jittery displays)
- ★ 48 chars by 16 lines — 1K memory mapped video system providing high speed access to screen display enabling animated games and graphs.
- ★ Extensive 256 character set which includes full upper and lower case alphanumerics. Greek symbols for mathematical constants and numerous graphic characters enabling you to form almost any shape you desire anywhere on the screen.
- ★ Video output and UHF Highgrade modulator (8Mz Bandwidth) which connects direct to the aerial socket of your T.V. Channel 36 UHF.
- ★ Fully stabilised 5V power supply including transformer on board.
- ★ Standard KANSAS city tape interface providing high reliability program storage — use on any standard domestic tape or cassette recorder.
- ★ 4K user RAM expandable to 8K on board £49 extra.
- ★ 40 line expansion interface socket on board for attachment of extender card containing 24K RAM and disk controller. (Ohio Scientific compatible).
- ★ 6502 machine code accessible through powerful 2K machine code monitor on board.
- ★ High quality thru plated P.C.B. with all I.C.'s mounted on sockets.
- ★ Professional 52 Key keyboard in 3 colours — software polled meaning that all debouncing and key decoding done in software.

COMMANDS

CONT LIST	NEW	NULL	RUN

STATEMENTS

CLEAR DATA	DEF	DIM	END	FOR
GOTO GOSUB	IF..GOTO	IF..THEN	INPUT	LET
NEXT ON..GOTO	ON..GOSUB	POKE	PRINT	READ
REM RESTORE	RETURN	STOP		

EXPRESSIONS

OPERATORS

+, -, *, /, ↑ NOT.AND.OR. >, <, <>, >=, <= RANGE 10^{-32} to 10^{+32}

VARIABLES

A.B.C. ....Z and two letter variables
The above can all be subscripted when used in an array. String variables use above names plus $.e.g.A$.

*8K Microsoft Basic means conversion to and from Pet, Apple and Sorcerer easy.
Many compatible programs already in print.

SPECIAL CHARACTERS

@ Erases line being typed, then provides carriage return, line feed.
← Erases last character typed.
CR Carriage Return — must be at the end of each line.
: Separates statements on a line.
CONTROL/C Execution or printing of a list is interrupted at the end of a line. "BREAK IN LINE XXXX" is printed, indicating line number of next statement to be executed or printed.
CONTROL/O No outputs occur until return made to command mode. If an Input statement is encountered, either another CONTROL/O is typed, or an error occurs.
? Equivalent to PRINT

Simple Soldering due to clear and consise instructions compiled by **Dr. A.A. Berk**, BSc.PhD

NO EXTRAS NEEDED JUST HIT 'RETURN' AND GO.

Build, understand, and program your own computer for only a small outlay.

KIT ONLY **£219** + VAT including RF Modulator & Power supply.
Absolutely no extras.

Available ready assembled and tested, ready to go for
£269 + VAT

FUNCTIONS

ABS(X)	ATN(X)	COS(X)	EXP(X)
LOG(X)	PEEK(I)	POS(I)	RND(X)
SPC(I)	SQR(X)	TAB(I)	TAN(X)
FRE(X)	INT(X)		
SGN(X)	SIN(X)		
USR(I)			

STRING FUNCTIONS

ASC(X$)	CHR$S(I)	FRE(X$)	LEFT$(X$.I)
RIGHT$(X$.I)		STR$(X)	
LEN(X$)	MID$(X$.I.J)		
VAL(X$)			

EXTRAS AVAILABLE SOON

COLOUR ADD-ON enables you to choose your foreground and background colour anywhere on the screen. Flash any character on the screen at will. Full documentation and parts in kit form.

AD-A-RAM EXTENDER CARD provides up to 32K Dynamic RAM Expansion, 8 Eprom sockets for 2708's or 2716's. Parallel Port (centronics compatible) and an RS232C serial port.

WIN YOURSELF AN ANADEX DP8000 LINE PRINTER

There's never enough good software around. That's why **COMPUKIT LTD.** are sponsoring a software contest. There are 2 categories:
1) **Business and Education**
2) **Fun and Games**
One lineprinter will be awarded to the winner of each category.
Send or bring along to the address shown below the following:
1) The program on cassette in the format used by the **COMPUKIT UK101**
2) Any documentation that you have for the program (source listing not necessary)
3) This coupon signed by you accepting the rules and conditions of the competition.

RULES:

1) Entries, including documentation, must be printed by computer or typed double spaced, with your name on every page.
2) Send or bring your entries to the address shown below.
3) Entries must be received by midnight on 29/2/80, any received after this time are void.
Winners will be notified by post before 31/3/80.
4) You warrant by your signature that all programs and documentation material included is entirely your own creation, and that no rights to it have been given or sold to any other party, and you agree to allow **COMPUKIT LTD.** to use, publish, distribute, modify, and edit it as it sees fit.
5) All entries become the property of **COMPUKIT LTD.** No entries will be returned nor any questions answered regarding individual entries.
6) Judging will be by a selected panel chosen by, and including representatives of COMPUKIT LTD. Judges may assign programs to any of the categories as they see fit. Decision of the judges is final.
7) Employees of **COMPUKIT LTD,** its dealers, distributors, advertising agencies and media are not eligible to enter.

Name ____________________

Address ____________________

____

____

I agree to abide by the above mentioned rules.

Signature ____________________

COMP COMPUTER COMPONENTS

14 STATION ROAD, NEW BARNET, HERTFORDSHIRE **TEL: 01-441 2922 (Sales)**
CLOSE TO NEW BARNET BR STATION — MOORGATE LINE **01-449 6596**
OPEN — 10am to 7pm — Monday to Saturday **TELEX: 298755**

All Products Ex-Stock Please check availability

(Part of the Compshop Ltd. Group).